



**TUGAS AKHIR - TF 145565**

**RANCANG BANGUN SISTEM PENGENDALIAN  
FLOWRATE BAHAN BAKAR GENERATOR SET  
*DUAL FUEL SYSTEM***

**RENANTYO ARDI PRAYOGO**  
NRP. 10511500000072

**DOSEN PEMBIMBING 1**  
**IR.HERI JOESTIONO ,MT**  
NIP . 19531116 198003 1 001  
**DOSEN PEMBIMBING 2**  
**AHMAD FAUZAN ADZIIIMAA ,ST.M.Sc**  
NIP . 1991201711052

**PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI**  
**DEPARTEMEN TEKNIK INSTRUMENTASI**  
**Fakultas Vokasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2018**



**FINAL PROJECT - TF 145565**

***DESIGN CONTROL SYSTEM THE FUEL  
FLOWRATE ON GENERATOR SET DUAL FUEL  
SYSTEM***

**RENANTYO ARDI PRAYOGO**

**NRP. 10511500000072**

**DOSEN PEMBIMBING 1**

**IR.HERI JOESTIONO ,MT**

**NIP . 19531116 198003 1 001**

**DOSEN PEMBIMBING 2**

**AHMAD FAUZAN ADZIIMAA ,ST.M.Sc**

**NIP . 19912017111052**

***STUDY PROGRAM OF D3 INSTRUMENTATION TECHNOLOGY***

***DEPARTMENT OF INSTRUMENTATION ENGINEERING***

***Faculty Of Vocation***

***Sepuluh November Institute of Technology***

***Surabaya 2018***

**LEMBAR PENGESAHAN I**

**“RANCANG BANGUN SISTEM PENGENDALIAN  
FLOWRATE BAHAN BAKAR GENERATOR SET  
DUAL FUEL SYSTEM ”**

**TUGAS AKHIR**

**Oleh :**

**RENANTYO ARDI PRAYOGO  
NRP.10511500000072**

**Surabaya, 19 Juli 2018  
Mengetahui / Menyetujui**

Dosen Pembimbing I



Ir. Heri Joestiono ,MT  
NIP. 19531116 198003 1 001

Dosen Pembimbing II



Ahmad Fauzan Adzimuza, ST, M.Sc  
NIP. 1991201711052



**Kepala Departemen  
Teknik Instrumentasi FV-ITS**  
  
Dr. Ir. Purwadi Agus Darwinto, M.Sc  
NIP. 19620822 198803 1 001

## LEMBAR PENGESAHAN II

### **“RANCANG BANGUN SISTEM PENGENDALIAN FLOWRATE BAHAN BAKAR GENERATOR SET DUAL FUEL SYSTEM ”**

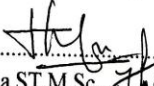



#### **TUGAS AKHIR**

Diajukan Untuk Memperoleh salah satu syarat  
Untuk memperoleh gelar Ahli Madya  
Pada  
Program studi D3 Teknologi Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

**Oleh :**

**RENANTYO ARDI PRAYOGO**  
**NRP.10511500000072**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Heri Joestiono ,MT..... (Pembimbing 1)
2. Ahmad Fauzan Adziimaa,ST,M.Sc..... (Pembimbing 2)
3. Dwi Oktavianto Wahyu N ST,MT..... (Penguji 1)
4. Arief Abdurrahman ST,MT..... (Penguji 2)

**SURABAYA**  
**2018**

# **RANCANG BANGUN SISTEM PENGENDALIAN FLOWRATE BAHAN BAKAR PADA GENERATOR SET DUAL FUEL SYSTEM**

**Nama** : Renantyo Ardi Prayogo  
**NRP** : 10511500000072  
**Departemen** : Teknik Instrumentasi-ITS  
**Dosen Pembimbing** : Ir. Heri Joestiono ,MT  
Ahmad Fauzan Adziimaa ,ST,M.Sc

## **ABSTRAK**

Energi alternatif diperlukan di era globalisasi ini karena energi dari dalam bumi yang ada saat ini sangat terbatas .Maka diperlukan teknologi untuk menemukan bahan yang sesuai yang ramah lingkungan dan lebih murah. Energi alternatif dapat dimanfaatkan dari berbagai aspek dan bahan. Sudah ditemukan banyak energi alternatif dengan harga murah yang menggunakan limbah dan kotoran.Oleh karena itu dirancang tugas akhir dengan judul rancang bangun pengendalian flowrate bahan bakar pada generator set dual fuel system.Jika tidak di kendalikan maka tidak akan didapatkan performansi genset yang maksimal.Pengendalian Flowrate menggunakan flowmeter hall effect YF-S402 dengan range pengukuran 0,3-6 L/min namun dalam proses kalibrasi range yang digunakan yaitu 1-5 L/min. Dengan Span 4 L/min , Akurasi sebesar 59,4%,Histerisis 0,009389671%, eror  $\pm 0,623922025$ . Pengendalian flowrate bahan bakar biogas dan bensin dilakukan dengan metode rasio yaitu perbandingan rasio biogas:bensin yaitu 90:10. Dengan asumsi bensin konstan pada flowrate 0,13 L/min sehingga flowrate biogas dijaga pada flowrate 1,17 L/min. Untuk mencapai set poin flowrate biogas membutuhkan waktu 35 detik dan mencapai posisi steady pada detik ke 100.

**Kata kunci** : Biogas, Genset, Pengendalian , Flowrate

# **DESIGN CONTROL SYSTEM THE FUEL FLOWRATE ON GENERATOR SET DUAL FUEL SYSTEM**

**Author's Name : Renantyo Ardi Prayogo**  
**NRP : 10511500000072**  
**Departement : Instrumentation Technology ITS**  
**Lecture : Ir. Heri Joestiono ,MT**  
**Ahmad Fauzan Adziimaa ,ST,M.Sc**

## ***ABSTRACT***

Alternative energy is needed in this era of globalization because the energy from within the earth is currently very limited and the longer the energy will be reduced in number. Recognizing alternative energy, research is needed to find suitable materials that are environmentally friendly and less expensive. Alternative energy can be exploited from various aspects and materials. Already found a lot of alternative energy at low prices that use waste and dirt. Therefore designed the final task with the title design of fuel flowrate control on the generator set dual fuel system. Flowrate control using flowmeter hall effect YF-S402 with measurement range 0.3 -6 L / min but in the calibration process the range used is 1-5 L / min. With Span 4 L / min, Accuracy of 59,4%, Hysteresis 0,009389671%, error  $\pm 0,623922025$ . Control of biogas and gasoline flowrate of fuel is done by ratio method that is ratio of biogas ratio: gasoline is 90:10. Assuming constant gasoline at a flowrate of 0.13 L / min so that the biogas flowrate is maintained at a flowrate of 1.17 L / min. To achieve the biogas flowrate set point takes 35 seconds and reach the steady position at 100 seconds.

**Keyword : Biogass, Genset, Control, Flowrate**

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas berkatNya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“RANCANG BANGUN PENGENDALIAN FLOWRATE BAHAN BAKAR PADA GENERATOR SET DUAL FUEL SYSTEM “** tepat pada waktunya.

Selama menyelesaikan tugas akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

- 1.Orang tua tercinta yang telah memberikan segala dukungan baik moral maupun materil serta dukungan yang sangat luar biasa
- 2.Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Kepala Departmen D3 Teknik Instrumentasi ITS yang telah memberikan semangat dan motivasi kepada kami
- 3.Bapak Arief Abdurrahman ST,MT. selaku dosen wali kami yang selalu membimbing dan memberikan motivasi kepada kami dalam menyelesaikan Tugas Akhir ini.
- 4.Bapak Ir. Heri Joestiono ,MT selaku dosen pembimbing pertama yang telah meluangkan waktu, arahan dan saran yang membantu selama proses pengerjaan tugas akhir.
- 5.Bapak Ahmad Fauzan Adziimaa ST,M.Sc selaku dosen pembimbing kedua yang telah meluangkan waktu, arahan dan saran yang membantu selama proses pengerjaan tugas akhir

6.Terimakasih kepada mbak dan mas serta alumni D3 Teknik Instrumentasi yang telah mendukung,membimbing, serta memberikan motivasi selama ini.

7.Terimakasih kepada BIOGAS PROJECT TEAM, Alif, Rifda, dan Tichu, yang bersama-sama berjuang dalam pengerjaan tugas akhir ini.

8.Teman-teman tercinta Workshop Intrumentasi 2015 yang selalu senantiasa memberikan semangat dan semua pihak yang telah membantu terselesaikannya Tugas Akhir ini.

9.Teman teman BHC (Bonek Heroes Campus) yang selalu mendukung serta menyemangati penulis dalam mengerjakan Tugas Akhir ini. Salam Satu Nyali, WANI

10.Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Surabaya, 19 Juli 2018

Renantyo Ardi Prayogo



## DAFTAR ISI

LEMBAR JUDUL.....	ii
LEMBAR PENGESAHAN I .....	<b>Error! Bookmark not defined.</b>
LEMBAR PENGESAHAN II.....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xixi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
BAB II TEORI PENUNJANG.....	5
2.1 Biogas sebagai energi alternatif .....	5
2.2 Teknologi Sistem Dual Fuel.....	6
2.3 Generator Set (Genset) .....	8
2.4 Mikrokontroler ATmega 128 .....	9
2.5 Flowmeter.....	11
2.6 Motor Operated Valve.....	11
2.7 Sistem Pengendalian .....	12
2.8 <i>Air Fuel Ratio</i> .....	14
2.9 Pengendalian Motor Servo pada Servo Operated Valve .....	14
BAB III PERANCANGAN DAN PEMBUATAN ALAT .....	17
3.1 Alat dan Bahan .....	17
3.2 <i>Flow chart/</i> Diagram Alir Pembuatan Alat .....	17

BAB IV HASIL DATA DAN PEMBAHASAN .....	25
4.1 Rancang Bangun Alat .....	25
4.2 Pengukuran Kalibrasi Sensor .....	26
4.3 Data Karakteristik Statik Sensor .....	30
4.4 Pengujian Alat.....	32
4.5 Pembahasan.....	39
BAB V PENUTUP.....	43
5.1 Kesimpulan .....	43
5.2 Saran.....	43
DAFTAR PUSTAKA	
LAMPIRAN	

## DAFTAR GAMBAR

Gambar 2.1 Plant biogas .....	6
Gambar 2.2 <i>Mixer Pada Teknologi Combustion Air Gas</i> <i>Integration</i> .....	7
Gambar 2.3 Generator Set <i>Dual Fuel Engine</i> .....	9
Gambar 2.4 Mikrokontroler ATmega128 .....	10
Gambar 2.5 Sensor flowmeter.....	11
Gambar 2.6 Motor Operated Valve.....	12
Gambar 2.7 Diagram Blok Sistem pengendalian.....	13
Gambar 2.8 Logika pengendalian motor servo.....	15
Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir.....	18
Gambar 3.2 P&ID dari Generator Set Dual Fuel System.....	20
Gambar 3.3 Diagram blok dari Sistem Pengendalian flowrate bahan bakar generator set <i>dual</i> <i>fuel system</i> .....	20
Gambar 3.4 Desain Plant Genset <i>Dual Fuel System</i> .....	21
Gambar 3.5 Desain Mekanik Plant Genset Dual Fuel Engine 3D .....	21
Gambar 3.6 Sesor Water Flowmeter yang digunakan .....	22
Gambar 3.7 Konfigurasi Pin ATmega128 .....	23
Gambar 3.8 Rangkaian Antarmuka RS232 .....	23
Gambar 3.9 Tampilan Program CV AVR .....	24
Gambar 4.1 Generator Set <i>Dual Fuel System</i> biogas bensin <i>Plant</i> .....	25
Gambar 4.2 Kalibrasi sensor flowmeter YF-S402 dengan Rotameter standar .....	27
Gambar 4.3 Grafik Linearitas Sensor .....	29
Gambar 4.4 Grafik Pembacaan Naik Sensor.....	31
Gambar 4.5 Grafik Pembacaan Turun Sensor.....	31
Gambar 4.6 Grafik respon aktuator.....	33
Gambar 4.7 Grafik Hasil Respon system rasio 90:10 .....	35
Gambar 4. 8 Grafik hasil respon sistem rasio 80:20 .....	37
Gambar 4.9 Grafik hasil respon sistem rasio 70:30.....	39

## DAFTAR TABEL

Tabel 2.1 Komponen penyusun biogas.....	5
Tabel 4.1 Tabel Pengujian Aktuator .....	33
Tabel 4.2 Hasil Pengujian Sistem rasio 90:10.....	34
Tabel 4.3 Hasil Pengujian Sistem rasio 80 : 20 .....	36
Tabel 4.4 Hasil Pengujian Sistem rasio 70 : 30.....	37

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kelangkaan bahan bakar minyak, yang disebabkan oleh kenaikan harga minyak dunia yang signifikan, telah mendorong pemerintah untuk mengajak masyarakat mengatasi masalah energi bersama-sama. Penghematan telah kita gerakkan sejak dahulu karena pasokan bahan bakar yang berasal dari minyak bumi adalah sumber energi fosil yang tidak dapat diperbarui, sedangkan permintaan naik terus, demikian pula harganya sehingga tidak ada stabilitas keseimbangan permintaan dan penawaran. Salah satu jalan untuk menghemat bahan bakar minyak (BBM) adalah mencari sumber energi alternatif yang dapat diperbarui.

Indonesia merupakan negara yang berpenduduk kelima di dunia memerlukan sumber pangan yang besar termasuk daging dan susu. Pemerintah sudah mencanangkan Indonesia menjadi negara yang swasembada daging pada tahun 2014 dan ada wacana untuk swasembada susu pada tahun 2020. Berbagai program telah dilakukan untuk dapat mencapai target tersebut termasuk peningkatan populasi ternak melalui proses pembibitan sapi lokal. Hal tersebut ternyata mendorong peningkatan populasi sapi dari tahun ke tahun. Pada tahun 2011 populasi sapi potong, sapi perah dan kerbau di Indonesia berturut-turut mencapai 597 ribu ekor sapi perah, 14,8 juta ekor sapi potong dan 1305 ekor kerbau (Pertanian, 2017). Total populasi sapi potong dan sapi perah di Indonesia pada tahun 2011 sudah mencapai 15.421.586 ekor (Statistik Peternakan, 2012). Apabila rata-rata per ekor mengeluarkan kotoran sebanyak 10 kg /hari maka di negara kita dihasilkan 154215,860 ton kotoran sapi setiap hari. Jika hanya 10 persen kotoran ternak jadi pupuk organik atau dimasukkan ke digester biogas maka sebanyak 138794,274 ton/hari kotoran sapi belum dimanfaatkan. Kotoran tersebut akan dibuang di selokan-selokan didepan rumah. Menurut UNDP, Korea Energy Management Corporation (KEMCO) dan PT. Bumi Harmoni Indraguna (2010), setiap 1 ekor ternak sapi/kerbau dapat

menghasilkan sekitar  $0,6\text{m}^3$  biogas per hari. Jumlah total sapi di Indonesia sekitar 15 juta ekor. Jumlah sapi tersebut berpotensi menghasilkan biogas sebesar 9 juta  $\text{m}^3/\text{hari}$ . Dalam sebulan potensi gas yang dihasilkan mencapai 270 juta  $\text{m}^3$  dan dalam setahun menghasilkan 3285 juta  $\text{m}^3$ . Potensi ekonomis biogas adalah sangat besar, hal tersebut mengingat bahwa 1  $\text{m}^3$  biogas dapat digunakan setara dengan 0,62 liter minyak tanah (Fahri,2011), maka produksi gas nasional tersebut di atas setara dengan 2,036 milyar liter minyak tanah. Jika biogas dimanfaatkan masyarakat Indonesia sebagai energi alternatif pengganti minyak tanah untuk memasak maka dapat menghemat dana sekitar Rp 22,4 trilyun/tahun (harga minyak tanah Rp 11.000,-).

Biogas dihasilkan dari proses anaerobik dan terdiri dari beberapa unsur. Komposisi biogas yang dihasilkan dari pencernaan anaerobik kurang lebih sekitar 60 - 70 %  $\text{CH}_4$ , sekitar 30 - 40%  $\text{CO}_2$ , kurang dari 1%  $\text{N}_2$ , dan sekitar 10 - 2000 ppm  $\text{H}_2\text{S}$  (Osorio & Torres, 2009). Gas metana menjadi unsur primer dan menjadi parameter utama dalam menentukan kualitas sebuah produk biogas.(Budiyono,2012)

Dalam penulisan tugas akhir ini akan dibahas tentang pengaplikasian mesin generator set dua bahan bakar (dual fuel) yaitu bensin dan biogas pada mesin generator set yang ditujukan untuk menghemat bahan bakar bensin dan meningkatkan efisiensi dari mesin generator set itu sendiri. Sebagaimana kita tahu bahwa pada mesin – mesin generator set konvensional, pemakaian bahan bakar bensin dan efisiensinya masih kurang maksimal.

Dari beberapa tinjauan tersebut, maka dalam Tugas Akhir ini dirancang sistem pengendalian aliran bahan bakar (bensin dan biogas) generator set dual fuel untuk mendapatkan efisiensi terbaik.

## 1.2 Rumusan Permasalahan

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana merancang sistem pengendalian *flowrate* bahan bakar (bensin dan biogas ) pada generator set *dual fuel system*?
2. Bagaimana menguji komposisi bensin dan biogas pada generator set *dual fuel system* ?
3. Bagaimana merancang sistem pengendalian Air Fuel Rasio (AFR) pada generator set *dual fuel system*?
4. Bagaimana menguji sistem pengendalian Air Fuel Rasio (AFR) pada generator set *dual fuel system*?

## 1.3 Batasan Masalah

Perlu diberikan beberapa batasan permasalahan agar Adapun batasan masalah ruang lingkup dari tugas akhir ini yaitu hanya membahas pengendalian aliran bensin serta biogas pada generaor set dua bahan bakar (*dual fuel*).

## 1.4 Tujuan

Berdasarkan latar belakang yang telah dijelaskan di atas didapatkan tujuan tugas akhir sebagai berikut :

1. Untuk merancang sistem pengendalian *flowrate* bahan bakar (bensin dan biogas ) pada generator set *dual fuel system*.
2. Untuk mengetahui hasil pengujian komposisi bensin dan biogas pada generator set *dual fuel system*.
3. Untuk merancang sistem pengendalian Air Fuel Rasio (AFR) pada generator set *dual fuel system*
4. Untuk mengetahui hasil pengujian sistem pengendalian Air Fuel Rasio (AFR) pada generator set *dual fuel system*.

### **1.5 Manfaat**

Adapun manfaat yang dapat diperoleh dari tugas akhir ini yaitu :

1. Tugas akhir ini dapat dijadikan untuk mengetahui pengaruh komposisi bensin serta biogas terhadap kinerja dari generator set *dual fuel system* (biogas dan bensin)
2. Tugas akhir ini dapat dijadikan sebagai ajang menambah pengetahuan bagi adik tingkat di program studi D3 Teknik Instrumentasi tentang sistem kontrol pada Generator Set serta Energi Terbarukan



## **BAB II**

### **TEORI PENUNJANG**

#### **2.1 Biogas sebagai energi alternatif**

Biogas merupakan sebuah proses produksi gas bio dari material organik dengan bantuan bakteri. Proses degradasi material organik ini tanpa melibatkan oksigen disebut anaerobic digestion gas yang dihasilkan sebagian besar (lebih 50%) berupa metana, material organik yang terkumpul pada digester (reaktor) akan diuraikan menjadi dua tahap dengan bantuan dua jenis bakteri. Tahap pertama material organik akan didegradasi menjadi asam lemak dengan bantuan bakteri pembentuk asam. Bakteri ini akan menguraikan sampah pada tingkat hidrolisis dan asidifikasi. Hidrolisis yaitu penguraian senyawa yang sederhana. Sedangkan asidifikasi yaitu pembentukan asam dari senyawa sederhana. (Rahayu, 2009)

**Tabel 2.1** Komponen penyusun biogas

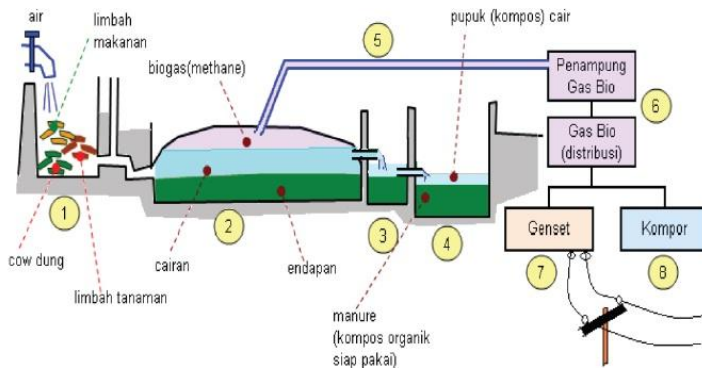
Komponen	Konsentrasi
Metana	50-75%
Karbon Dioksida	25-45%
Air	2-7%
Hidrogen Sulfide	20-20.000 ppm
Nitrogen	< 2%
Oksigen	< 2%
Hidrogen	< 1%

Setelah material organik berubah menjadi asam lemak, maka tahap kedua dari proses anaerobik digestion adalah pembentukan gas metana dengan bantuan bakteri pembentuk metana seperti *methanococcus*, *methanosarcina*, *methanobacterium*.

Biogas sebagian besar mengandung gas metana ( $\text{CH}_4$ ) dan karbon dioksida ( $\text{CO}_2$ ), dan beberapa kandungan yang jumlahnya

kecil diantaranya hydrogen sulfide ( $H_2S$ ) dan ammonia ( $NH_3$ ) serta hydrogen dan ( $H_2$ ), nitrogen sulphur, kandungan air dan karbon dioksida ( $CO_2$ ).

Energi yang terkandung dalam biogas tergantung dari konsentrasi metana ( $CH_4$ ). Semakin tinggi kandungan metana maka semakin besar kandungan energi (nilai kalor) pada biogas, dan sebaliknya semakin kecil kandungan metana ( $CH_4$ ) semakin kecil nilai kalor. Kualitas biogas dapat ditingkatkan dengan memperlakukan parameter yaitu : menghilangkan hydrogen sulphur, kandungan air dan karbon dioksida ( $CO_2$ ) (Wahyuni,2013)



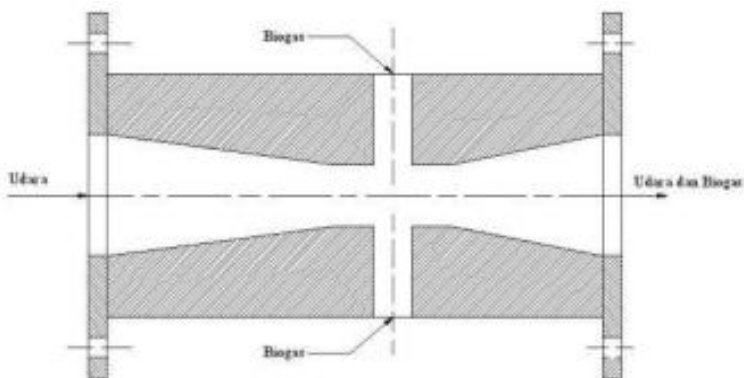
Gambar 2.1 Plant biogas

## 2.2 Teknologi Sistem Dual Fuel

Bahan bakar gas ini dicampur dengan udara dalam silinder mesin baik melalui pencampuran langsung di intake manifold dengan udara atau melalui suntikan langsung ke dalam silinder. Sebuah mesin dual fuel pada dasarnya adalah mesin diesel yang dimodifikasi di mana bahan bakar gas, disebut bahan bakar utama, yang dicampur bersama dengan udara dengan menggunakan mixing yang berbentuk venturi yang akan masuk melalui intake manifold. Bahan bakar ini adalah sumber utama energi input ke mesin. Bahan bakar gas utama dikompresi dengan

udara, bahan bakar gas memiliki temperatur terbakar sendiri lebih tinggi dibandingkan minyak solar. Sedangkan bahan bakar diesel, biasanya disebut pilot fuel, di injeksi seperti pada mesin diesel biasa di dekat akhir kompresi primer campuran bahan bakar udara. Bahan bakar pilot diesel merupakan yang melakukan pengapian pertama dan bertindak sebagai sumber pengapian untuk pembakaran dari campuran bahan bakar udara gas. Bahan bakar pilot diesel, yang diinjeksi ke ruang bakar hanya menyumbang sebagian kecil dari tenaga mesin yang dihasilkan. *Combustion Air Gas Integration* merupakan teknologi *dual-fuel diesel engine* yang memiliki dua sistem penyuplai bahan bakar yang berbeda. Dalam sistem ini bahan bakar gas disebut sebagai bahan bakar primer dan bahan bakar minyak solar disebut sebagai bahan bakar sekunder yang bertindak sebagai *pilot fuel*.

Pada sistem bahan bakar tunggal, mesin diesel bekerja dengan menghisap udara murni dari luar. Akan tetapi untuk sistem dual-fuel, yang masuk ke dalam ruang bakar tidak hanya udara murni. Udara tersebut akan masuk bersamaan dengan sejumlah bahan bakar gas.



Gambar 2.2 *Mixer Pada Teknologi Combustion Air Gas Integration*

Mekanisme pencampuran udara dan bahan bakar gas tersebut dapat terjadi di dalam *mixer* yang berbentuk yang

berbentuk venturi, dengan menggunakan injektor gas yang bertekanan tinggi yang langsung menginjeksikan gas ke ruang bakar, serta menggunakan injektor gas bertekanan rendah yang diletakkan di mulut saluran isap dan hanya akan menginjeksikan gas pada saat katup isap terbuka. (Osorio,2009)

### 2.3 Generator Set (Genset)

Genset atau yang merupakan singkatan dari generator set ini adalah sebuah perangkat yang mampu menghasilkan daya listrik. Genset ini merupakan seperangkat atau gabungan antara generator atau alternator dan *engine* yang dapat digunakan sebagai alat pembangkit listrik. Genset mampu digunakan sebagai sistem cadangan listrik atau "*off-grid*" (sumber daya yang tergantung atas kebutuhan pemakai). Genset dipakai oleh rumah sakit dan industri yang menginginkan sumber listrik yang besar dan relatif stabil. Generator terpasang satu poros dengan motor diesel, yang biasanya memakai generator sinkron (alternator) pada pembangkitan. Generator sinkron mempunyai dua bagian utama yaitu sistem medan magnet dan jangkar. Generator ini kapasitasnya besar, medan magnetnya berputar karena terletak pada rotor. Genset bekerja sepuluh detik ketika listrik padam, sepuluh detik berikutnya tenaga listrik diswitch ke genset, saat itu lampu bisa menyala kembali. Cara kerja generator genset yang memberikan supply listrik setelah duapuluh detik ini ditopang oleh AVR (*Automatic Voltage Regulator*). Di dalam AVR, ada *Mutual Reactor* (MT) yaitu semacam trafo jenis CT (*Current Transformer*) yang menghasilkan arus listrik berdasarkan besaran arus beban yang melaluinya (secara rangkaian seri). Arus listrik yang dihasilkan ini digunakan untuk memperkuat medan magnet pada belitan rotor. Sehingga untuk beban yang besar, arus yang dihasilkan juga besar  $V=I \times R$ , dimana  $V_p/V_s=I_p/V_p$  dan  $P=I \times V$ . (Artayana,2014)



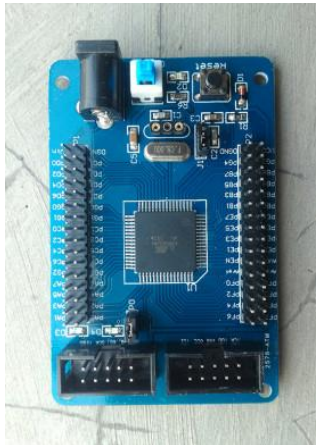
Gambar 2.3 Generator Set *Dual Fuel Engine*

## 2.4 Mikrokontroler ATmega128

Mikrokontroler ATmega128 merupakan salah satu varian dari mikrokontroler AVR 8-bit. Beberapa fitur yang dimiliki adalah memiliki beberapa memory yang bersifat non-volatile, yaitu 128Kbytes of In-System Self-Programmable Flash program memory (128Kbytes memory flash untuk pemrograman), 4Kbytes memori EEPROM, 4Kbytes memori Internal SRAM, write/erase cycles : 10.000 Flash/100.000 EEPROM (program dalam mikrokontroler dapat diisi dan dihapus berulang kali sampai 10.000 kali untuk flash memori atau 100.000 kali untuk penyimpanan program/data di EEPROM).

Selain memory, fitur yang dimiliki oleh mikrokontroler atmega128 ini adalah pada perangkat peripheral interfacenya, yaitu memiliki 2 buah 8-bit Timer/Counter, 2 buah expand 16-bit Timer/Counter, RTC (Real Time Counter) dengan oscillator yang terpisah, 2 buah 8-bit channel PWM, 6 PWM channel dengan resolusi pemrograman dari 2 sampai 16 bits, output compare modulator, 8-channel 10-bit ADC, 2 buah TWI (Two Wire Interface), 2 buah serial USARTs, Master/Slave SPI serial interface, Programmable Watchdog Timer dengan On-chip Oscillator, On-chip analog comparator, dan memiliki 53 programmable I/O. Sedangkan untuk pengoperasiannya sendiri, Mikrokontroler ATmega128 dapat dioperasikan pada catuan 2.7

- 5.5 V untuk ATmega128L (low voltage) dengan clock speed 0 – 8 MHz dan 4.5 – 5.5 V untuk ATmega128 dengan clock speed 0 – 16 MHz. (AtMEL,2005)



Gambar 2.4 Mikrokontroler ATmega128

Sistem minimum merupakan suatu rangkaian minimalis yang dirancang / dibuat agar suatu mikrokontroler dapat berfungsi dan bekerja dengan semestinya. Sama seperti mikrokontroler atmega8535, atmega128 juga membutuhkan sistem minimum, Namun sistem minimum pada Mikrokontroler ATmega128 memiliki beberapa perbedaan dibandingkan dengan sistem minimum mikrokontroler keluarga AVR yang lain. Perbedaan terletak pada konfigurasi pin pada ISP (In System Programming). Jika pada kebanyakan mikrokontroler jenis AVR konfigurasi pin untuk ISP-nya adalah mosi-mosi, miso-miso, sck-sck, reset-reset, dan power supply, maka pada Mikrokontroler ATmega128 adalah mosi-RX0, miso-TX0, SCK-SCK, dan power supply.

## 2.5 Flowmeter

Water Flow sensor terdiri dari tubuh katup plastik, rotor air, dan sensor hall efek. Ketika air mengalir melalui, gulungan rotor-rotor. Kecepatan perubahan dengan tingkat yang berbeda aliran. Sesuai sensor hall efek output sinyal pulsa. Water flow sensor ini terdiri atas katup plastik, rotor air, dan sebuah sensor hall-effect. Prinsip kerja sensor ini adalah dengan memanfaatkan fenomena efek Hall. Efek Hall ini didasarkan pada efek medan magnetik terhadap partikel bermuatan yang bergerak. Ketika ada arus listrik yang mengalir pada divais efek Hall yang ditempatkan dalam medan magnet yang arahnya tegak lurus arus listrik, pergerakan pembawa muatan akan berbelok ke salah satu sisi dan menghasilkan medan listrik. Medan listrik terus membesar hingga gaya Lorentz yang bekerja pada partikel menjadi nol. Perbedaan potensial antara kedua sisi divais tersebut disebut potensial Hall. Potensial Hall ini sebanding dengan medan magnet dan arus listrik yang melalui *device*. (Restiya Maulana, 2016)



Gambar 2.5 Sensor flowmeter

## 2.6 Motor Operated Valve

Motor Operated Valve merupakan ball valve yang dikendalikan dengan motor baik motor AC maupun DC tergantung motor yang digunakan. Motor valve on-off ini merupakan elemen aktuatur yang sering digunakan dalam sistem fluida. Seperti pada sistem pneumatik, sistem hidrolik

ataupun sistem kontrol fluida yang membutuhkan elemen penggerak otomatis. Contohnya pada sistem pneumatik, Motor Operated Valve bertugas untuk mengontrol saluran udara bertekanan yang menuju aktuator pneumatik . Atau pada tandon air yang dikontrol level ketinggian dari tandon tersebut.



Gambar 2.6 Motor Operated Valve

## 2.7 Sistem Pengendalian

Sistem pengendalian adalah susunan komponen komponen fisik yang dirakit sedemikian rupa sehingga berfungsi untuk mengendalikan sistem itu sendiri atau sistem lain yang berhubungan dengan sebuah proses. Atau dengan pengertian lain sistem pengendalian adalah suatu proses/pengendalian terhadap beberapa besaran sehingga berada pada suatu harga atau range tertentu. Hampir semua proses dalam dunia industri membutuhkan peralatan-peralatan otomatis untuk mengendalikan parameter– parameter prosesnya. Otomatis tidak saja diperlukan demi kelancaran operasi, keamanan, ekonomi, maupun mutu produk, tetapi lebih merupakan kebutuhan pokok.<sup>[13]</sup> Kita tidak akan mungkin menjalankan suatu proses industri tanpa sistem pengendalian, contohnya pengendalian disuatu proses pengilangan minyak. Ada banyak parameter yang harus dikendalikan di dalam suatu proses. Di antaranya yang paling umum adalah tekanan (*pressure*) di dalam sebuah vassel atau pipa, aliran (*flow*) di dalam pipa, suhu (*temperature*) di unit proses seperti *heat exchanger*, atau permukaan zat cair (*level*)

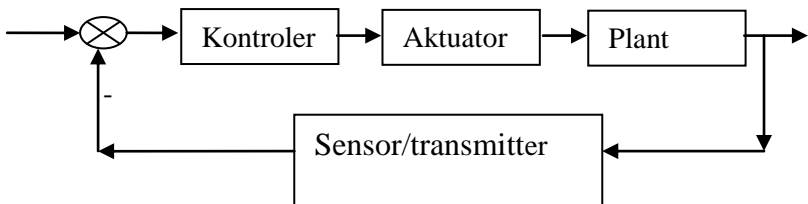


disebuah tangki.<sup>[2]</sup> Dan ada beberapa parameter lain diluar keempat parameter diatas yang cukup penting dan juga perlu dikendalikan karena kebutuhan spesifik proses, diantaranya pH di industri petrokimia, *water cut* (BS & W) di ladang minyak mentah, warna produk di suatu fasilitas pencairan gas (NGL) dan sebagainya. Gabungan serta kerja alat-alat pengendalian otomatis itulah yang dinamakan sistem pengendalian proses (*process control system*). Sedangkan semua peralatan yang membentuk sistem pengendalian disebut instrumentasi pengendalian proses (*process control instrumentation*). Kedua hal tersebut berhubungan satu sama lain, namun keduanya mempunyai hakikat yang berbeda. Ilmu process control instrumentasi lebih terfokus pada penjelasan kerja alat sedangkan process kontrol sistem lebih terpusat pada kerja sistem tersebut.

Pada akhirnya pengendalian otomatis memegang peranan penting dan memberikan kemudahan dalam mendapatkan performansi dalam suatu sistem dinamik, mempertinggi kualitas, menurunkan biaya produksi dan mempertinggi laju produksi, serta meniadakan pekerjaan pekerjaan rutiin yang harus dilakukan oleh manusia.

Namun semua peran operator manual digantikan oleh sebuah alat yang disebut *controller*. Tugas membuka dan menutup *valve* tidak lagi dikerjakan oleh operator tetapi atas perintah *controller*.

Untuk keperluan pengendalian otomatis, *valve* harus dilengkapi dengan alat yang disebut *actuator* , sehingga unit *valve* yang sekarang menjadi unit yang disebut *control valve*. Semua peralatan pengendalian inilah (*controller dan control valve*) yang disebut sebagai instrumentasi pengendalian proses. (XVII, 2007)



Gambar 2.7 Diagram Blok Sistem pengendalian

## 2.8 *Air Fuel Ratio*

Air Fuel Ratio adalah faktor yang mempengaruhi kesempurnaan proses pembakaran di dalam ruang bakar. Merupakan komposisi campuran bensin dan udara . Idealnya AFR bernilai 14,7 . Artinya campuran terdiri dari 1 bensin berbanding 14,7 udara atau disebut dengan istilah Stoichiometry. Pemakaian udara yang tidak stoikiometris, dikenal istilah Equivalent Ratio (ER). Equivalent Ratio (ER) adalah perbandingan antara jumlah (bahan bakar/ udara) yang digunakan dan jumlah (bahan bakar/ udara) stoikiometris.

Pada umumnya perbandingan udara dan bahan bakar dinyatakan berdasarkan perbandingan berat udara dengan berat bahan bakar. Perbandingan udara dan bahan bakar yang sempurna atau *air fuel ratio* (AFR) adalah 14,7 : 1, yaitu 14,7 udara berbanding 1 bensin. Tetapi pada praktiknya, mesin membutuhkan campuran udara dan bahan bakar dalam perbandingan yang berbeda-beda. Ini bergantung pada temperatur, kecepatan mesin dan kondisi lainnya. ( Wisnu Arya Wardana, 2001)

## 2.9 **Pengendalian Motor Servo pada Servo Operated Valve**

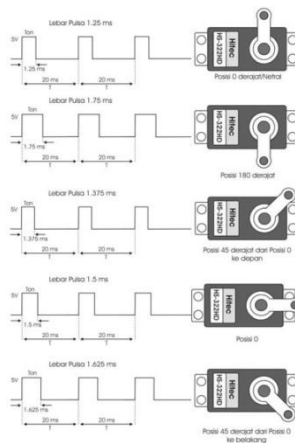
Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem closed feedback yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo.

Motor servo disusun dari sebuah motor DC, gearbox, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo. Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya

dapat dikendalikan dengan memberikan variasi lebar pulsa (duty cycle) sinyal PWM pada bagian pin kontrolnya.

Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0°/ netral).

Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton duty cycle, dan bertahan diposisi tersebut.



Gambar 2.8 Logika pengendalian motor servo

*Halaman ini sengaja dikosongkan*

## **BAB III**

### **PERANCANGAN DAN PEMBUATAN ALAT**

#### **3.1 Alat dan Bahan**

Pada tugas akhir alat dan bahan yang digunakan terdiri atas perangkat lunak dan perangkat keras sebagai berikut :

##### **Alat (Perangkat Keras)**

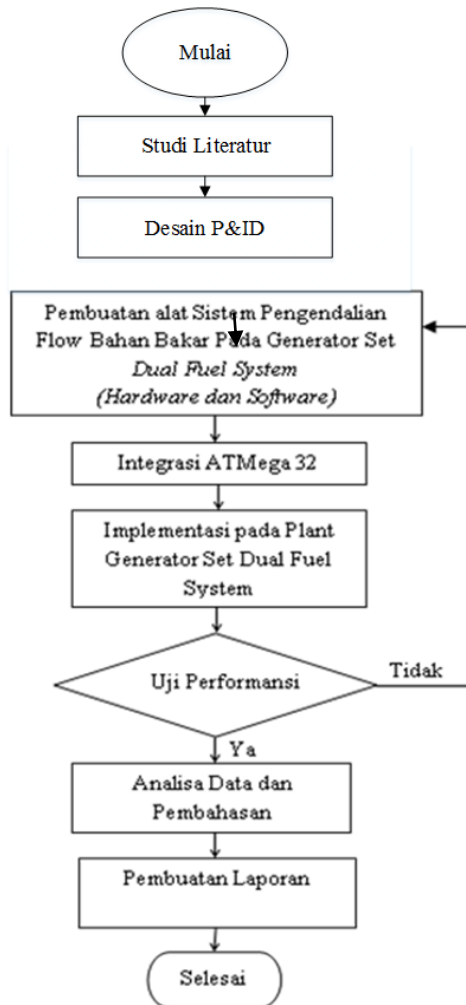
- a. Display LCD 16x4
- b. Mikrokontroler ATmega 32
- c. Port RS232
- d. Sensor *water flowmeter*
- e. Power supply 12 V DC
- f. Kabel Jumper
- g. Kabel Downloader ATmega
- h. PCB
- i. Servo Operated Valve
- j. Mini Circuit Breaker
- k. Kabel Power
- l. Adaptor 5 V DC

##### **Bahan (Perangkat Lunak)**

- a. Code vision AVR
- b. AVR Loader
- c. Proteus
- d. AutoCad Design (P&ID)
- e. Sketchup
- f. Microsoft Visio

#### **3.2 *Flow chart/ Diagram Alir Pembuatan Alat***

Pada subbab ini dijelaskan mengenai prosedur tahapan dalam penelitian tugas akhir yang dilakukan. Dengan *flowchart* dibawah ini :



Gambar 3.2 Diagram Alir Pengerjaan Tugas Akhir

Keterangan **Gambar 3.1** mengenai diagram alir dapat dijelaskan sebagai berikut :

**a. Studi Literature**

Dilakukan kajian terhadap metode-metode, konsep, atau teori yang terkait dengan tugas akhir yang dilakukan, baik yang bersumber dari jurnal, laporan, penelitian maupun buku yang memiliki bahasan yang sesuai dengan tema tugas akhir

**b. Perancangan Sistem Pengendalian**

Dilakukan perancangan sistem pengendalian aliran bensin dan biogas dengan membuat desain mekanik yang akan digunakan .

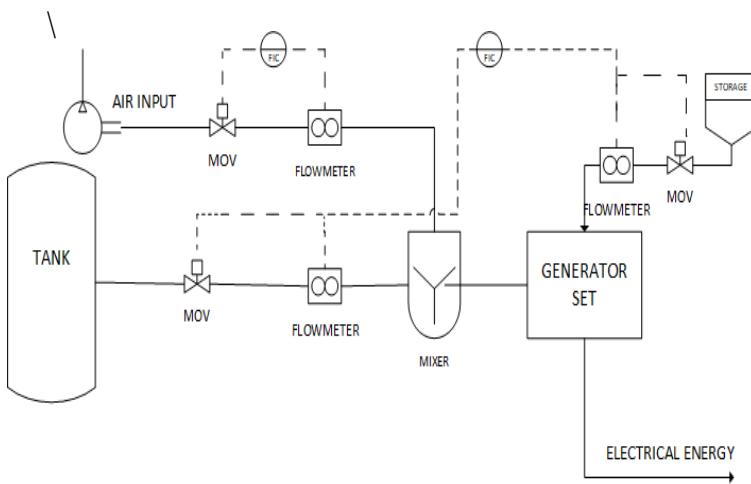
**c. Pembuatan Alat**

Dilakukan pembuatan sistem monitoring pada kecepatan rpm, pembuatan sistem monitoring dilakukan dengan membuat *hardware* dan *software* dari sistem monitoring. Pembuatan *hardware* meliputi pembuatan mekanik alat dan pembuatan *software* meliputi pembuatan visualisasi data yang akan ditampilkan berdasarkan hasil dari alat ukur

**d. Analisis Kinerja Sistem Pengendalian**

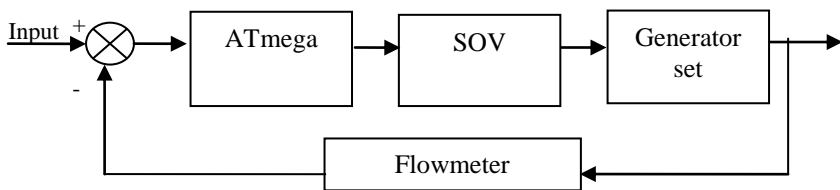
Analisis bertujuan untuk mengetahui apakah kinerja dari masing-masing komponen sesuai dengan perancangan mekanik alat serta bentuk visualisasi dari hasil pengukuran, apabila tidak sesuai maka akan dilakukan evaluasi terhadap perancangan sistem pengendalian flow bahan bakar pada generator set.

Adapun desain P&ID dari Tugas Akhir ini adalah sebagai berikut:



Gambar 3.2 P&ID dari Generator Set Dual Fuel System

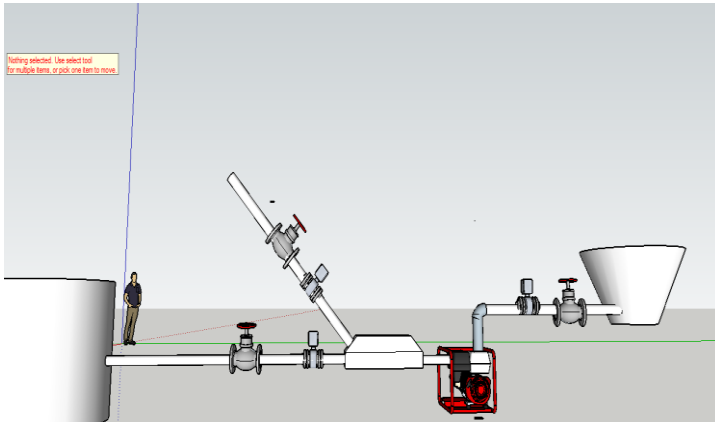
Adapun diagram blok dari Tugas Akhir ini adalah sebagai berikut:



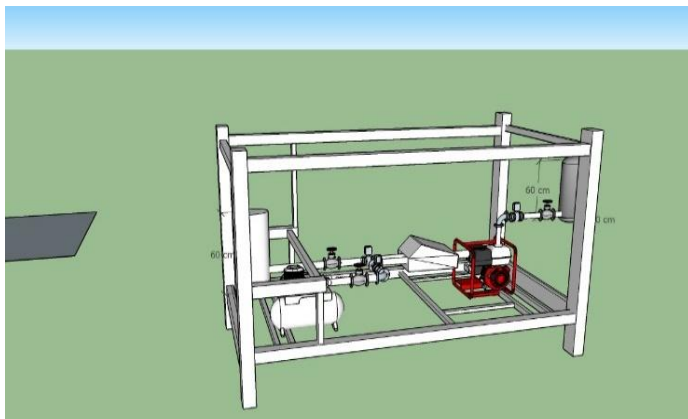
Gambar 3.3 Diagram blok dari Sistem Pengendalian flowrate bahan bakar generator set *dual fuel system*.



Pada pembuatan tugas akhir ini rancangan alat adalah sebagai berikut :



Gambar 3.4 Desain Plant Genset *Dual Fuel System*



Gambar 3.5 Desain Mekanik Plant Genset Dual Fuel Engine 3D

### 3.2.1 Perancangan *Hardware*

Pembuatan *hardware* untuk sistem monitoring ini terdapat perancangan pada rangkaian *sensing element*, perancangan pada mikrokontroler ATmega128, dan display pada LCD berikut ini merupakan penjelasan setiap perancangan tersebut :

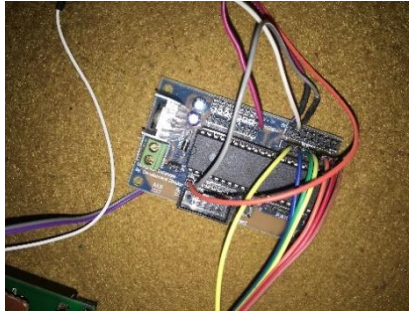
#### **Perancangan rangkaian *sensing element***

Dalam perancangan tugas akhir ini, sensor yang digunakan adalah sensor Water Flowmeter. Sensor ini diletakkan pada piping biogas serta bensin.



Gambar 3.6 Sesor Water Flowmeter yang digunakan

Sebagai aktivasi mikrokontroler ATmega128 ini mendapatkan sumber tegangan 5 volt. Untuk koneksi dari tiap pin terhadap instrumen digunakan kabel *jumper*, Dapat dilihat pada gambar berikut :



Gambar 3.7 Konfigurasi Pin ATmega128

### **Rangkaian Antarmuka RS232**

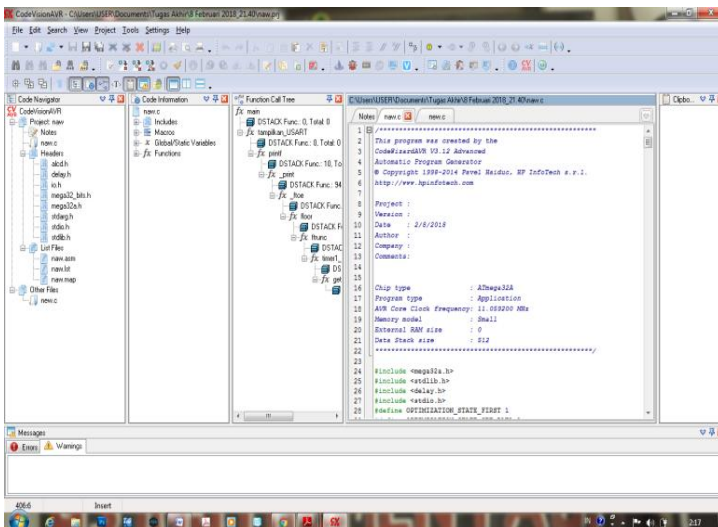
RS232 adalah *standard* komunikasi serial yang digunakan untuk koneksi periperal ke periperal. Biasa juga disebut dengan jalur I/O ( *input/output* ). Fungsi dari serial port RS232 adalah untuk menghubungkan/koneksi dari perangkat yang satu dengan perangkat yang lain, atau peralatan standart yang menyangkut komunikasi data antara komputer dengan alat-alat pelengkap komputer.



Gambar 3.8 Rangkaian Antarmuka RS232

### 3.2.2 Perancangan *software*

Pada pembuatan tugas akhir ini dibuat *software* penunjang untuk sistem monitoring, *software* yang digunakan merupakan CodeVisionAVR sebagai *interface* dari ATmega128. Pada *software* pemrograman CodeVisionAVR dilakukan pemrograman LCD, ADC dan RS232 (komunikasi serial), kemudian rangkaian *sensing element* dalam pemrograman CodeVisionAVR diintegrasikan dengan men-download program CodeVisionAVR ke mikroontroller ATmega128, kemudian akan diproses pada rangkaian pemrosesan sinyal dan akan ditampilkan ke LCD pada panel dan PC dalam bentuk data *real time* waktu konsumsi bensin dengan menggunakan komunikasi serial RS232 sebagai koneksi antarmuka antara pemrosesan sinyal mikrokontroller ATmega128 dengan PC.

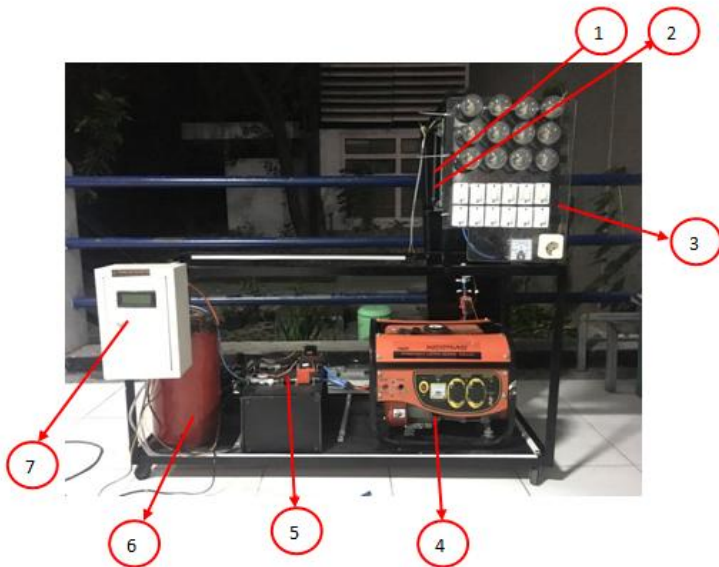


Gambar 3.9 Tampilan Program CV AVR

## BAB IV HASIL DATA DAN PEMBAHASAN

### 4.1 Rancang Bangun Alat

Berikut ini adalah perancangan sistem pengendalian *flow* bahan bakar pada generator set *dual fuel system*.



Keterangan :

1. Tangki Storage Bensin
2. Kontrol Flow bensin
3. Papan Uji beban
4. Generator set dual fuel
5. Kontrol Flow udara dan biogas
6. Tangki storage biogas
7. Panel box

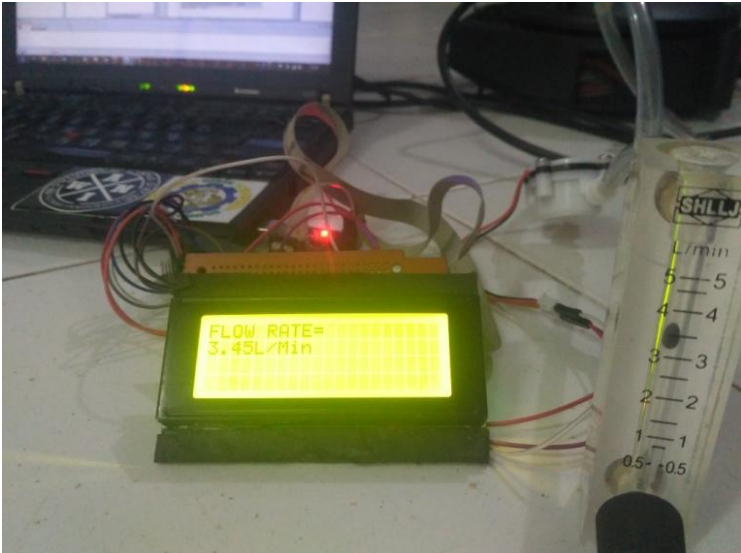
Gambar 4.1 Generator Set *Dual Fuel System* biogas  
bensin *Plant*

Dari hasil biodigester biogas, maka gas akan diteruskan pada purifikasi untuk penghilangan gas pengotor lainnya yang dapat menurunkan kualitas dari biogas tersebut. Setelah melalui proses purifikasi maka biogas akan dimanfaatkan sebagai bahan bakar untuk pembangkit listrik yang dalam hal ini adalah generator set dual fuel system yaitu bahan bakar biogas-bensin.

Agar didapatkan tingkat efektifitas dan efisiensi yang lebih tinggi maka dilakukan sistem control flow pada biogas dan bensin, hal ini dapat berpengaruh pada segi ekonomis serta performa dari genset *dual fuel system*. Proses control flow bahan bakar biogas dan bensin ini menggunakan metode rasio, yaitu dengan mengatur komposisi antara biogas serta bensin yang masuk pada genset. Dalam Tugas Akhir ini digunakan rasio 90:10, 80:20, dan 70:30 antara biogas:bensin. Pada proses ini menggunakan sensor flowmeter dengan aktuator berupa servo operated valve yang dapat mengendalikan aliran bensin serta biogas yang masuk pada sistem pembakaran genset dual fuel system dengan setpoint komposisi antara biogas dan bensin sesuai dengan yang diatur pada layar monitor HMI pada panel box.

## 4.2 Pengukuran Kalibrasi Sensor

Secara umum kalibrasi mempunyai pengertian sebagai rangkaian kegiatan membandingkan hasil pengukuran suatu alat dengan alat standar yang sesuai untuk menentukan besarnya koreksi pengukuran alat serta ketidakpastiannya. Kalibrasi sensor ini sangat diperlukan agar keluaran sensor dapat sesuai dengan yang sudah ditetapkan atau standart. Kalibrasi flowmeter YF-S402 ini menggunakan rotameter sebagai alat pembanding yang sudah sesuai standar. Pembacaan rotameter dibandingkan dengan LCD yang menampilkan hasil pembacaan dari flowmeter YF-S402.



Gambar 4.2 Kalibrasi sensor flowmeter YF-S402 dengan Rotameter standar

Pada Kalibrasi sensor flowmeter YF-S402 ini dilakukan dengan range pengukuran 1-5 L/min , dengan masing-masing 5 kali pembacaan setiap satuan dan dilakukan pengukuran naik dan turun.

Berikut ini merupakan cara untuk mencari nilai ketidakpastian sensor Flow meter YF-S402

a. Nilai Ketidakpastian Tipe A

$$\sigma = \frac{\sqrt{\sum(y-\bar{y})^2}}{n-1} \dots\dots\dots (4.1)$$

$$\sigma = 0,259057518$$

Sehingga didapatkan nilai  $U_{a1}$  sebagai berikut:

$$U_{a1} = \frac{\sigma}{\sqrt{n}} \dots\dots\dots (4.2)$$

$$U_{a1} = 0,11616929$$

Sedangkan nilai ketidakpastian regresi  $U_{a2} = \sqrt{\frac{SSR}{n-2}}$

Dimana:

$$SSR \text{ (Sum Square Residual)} = \sum SR \text{ (Square Residual)}$$

$$SR = R^2 \text{ (Residu)}$$

$$Y = \text{Nilai koreksi}$$

$$Y_{reg} = a + (b \cdot x_i) \dots\dots\dots (4.3)$$

$$a = \bar{y}_l + (b \cdot \bar{x}) \dots\dots\dots (4.4)$$

$$b = \frac{n \cdot \sum xy - \sum x \sum y}{n \cdot \sum x^2 - (\sum x)^2} \dots\dots\dots (4.5)$$

$$SSR = \sum (y - Y_{reg})^2 \dots\dots\dots (4.6)$$

Dimana:

$$y = \text{Nilai koreksi,}$$

$$n = \text{Jumlah data}$$

Sehingga nilai:

$$a = (-0,138) + (-0,29314 \times 2,5)$$

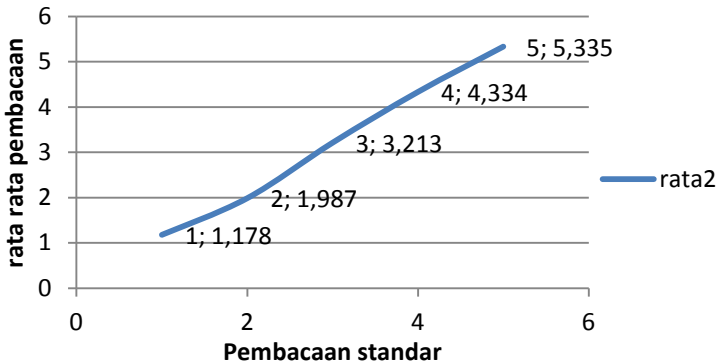
$$a = -0,9$$

Jadi, persamaan regresi menjadi:

$$Y_{reg} = (-0,9) + (x(-0,29314))$$



## Linearitas



Gambar 4.3 Grafik Linearitas Sensor

Yang menghasilkan nilai **SSR** =

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{12,68}{5-2}} = 2,056$$

b. Nilai Ketidakpastian *Type B* :

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi ( $U_{b1}$ ) dan Ketidakpastian alat standar *pressure gauge* ( $U_{b2}$ ). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{\frac{1}{2} \times \text{Resolusi}}{\sqrt{3}} = \frac{\frac{1}{2} \times 1}{\sqrt{3}} = 0,288$$

$U_{b2} = \frac{a}{k}$ , dikarenakan pada alat standar tidak ada sertifikat kalibrasinya maka nilai  $a$  (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil :  $U_{b2} = 0$

- c. Nilai ketidakpastian kombinasi  $U_c$ :

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \dots\dots\dots(4.7)$$

$$U_c = 1,281749834$$

#### 4.3 Data Karakteristik Statik Sensor

Karakteristik statik merupakan sifat dari sebuah alat ukur yang tidak bergantung terhadap waktu. Berikut merupakan data perhitungan. Untuk mengetahui nilai untuk mengetahui karakteristik suatu alat.

Dari data yang terdapat dalam tabel maka dapat diketahui nilai karakteristik alat sebagai berikut:

- |    |            |                     |
|----|------------|---------------------|
| a. | Range      | : 1-5 L/min         |
| b. | Span       | : 4                 |
| c. | Histerisis | : 0,009389671%      |
| d. | Akurasi    | : 59,4%             |
| e. | Error      | : $\pm 0,623922025$ |

Nilai tersebut dapat diketahui dengan menghitung menggunakan rumusan seperti berikut:

- a. Histerisis

$$\text{Histerisis } H(I) = O(I)_{I\downarrow} - O(I)_{I\uparrow}, \hat{H} = H(I)_{max}$$

sehingga:

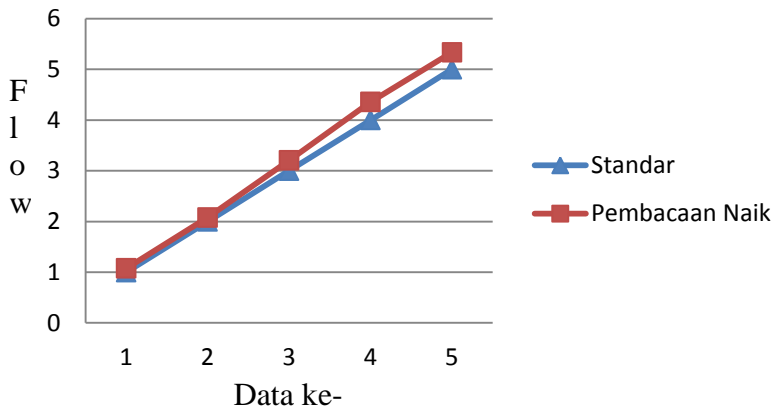
$$\% \text{ Maksimum histeresis} = \frac{\hat{H}}{O_{max} - O_{min}} \times 100\%$$

$$\% \text{ Maksimum histeresis} = \frac{0,0004}{5,336 - 1,076} \times 100 \%$$

$$\% \text{ Maksimum histerisis} = 0,009389671\%$$

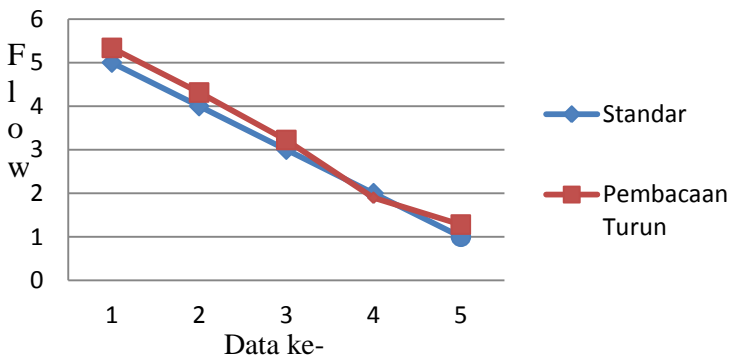
Berikut gambar grafik histerisis dari data pengukuran naik dan turun tersebut:

### Pembacaan Naik Sensor



Gambar 4.4 Grafik Pembacaan Naik Sensor

### Pembacaan Turun Sensor



Gambar 4.5 Grafik Pembacaan Turun Sensor

Pada gambar grafik diatas menunjukkan bahwa histerisis tertinggi terletak pada saat output 5,336 L/m.

b. Akurasi

$$A = 1 - \Sigma \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%$$

Dengan:

$Y_n$  = Pembacaan standar (I) dan

$X_n$  = Pembacaan alat (O)

A = 59,4%

c. Error

$$\text{Error} = \pm (1,96\sigma + \alpha)$$

$\sigma$  = standar deviasi

$$\alpha = \sigma / \sqrt{n}$$

$$\text{Error} = \pm 0,623922025$$

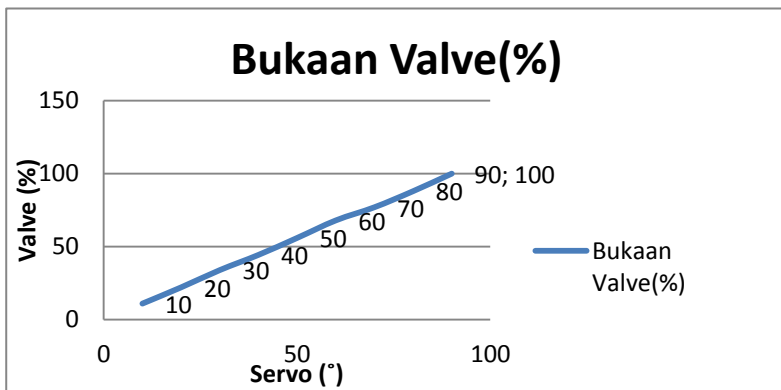
#### 4.4 Kalibrasi Aktator

Aktuator sangat penting dalam sistem pengendalian, dalam tugas akhir ini digunakan servo operated valve , yaitu sebuah valve yang di couple dengan motor servo. Sedangkan tipe dari valve yang digunakan yaitu ball valve yang hanya dapat membuka dan menutup dengan putaran 90°. Sedangkan pada motor servo dapat bergerak dengan 180°, dengan pembagian 90° ke kanan dan 90° ke kiri. Dari pergerakan motor servo tersebut maka akan dikonversi ke dalam bukaan valve. Maka akan dihasilkan data pengujian aktuator sebagai berikut :

**Tabel 4.1** Tabel Pengujian Aktuator

Motor servo( °)	Bukaan Valve(%)
10	11,1
20	22,2
30	34
40	44,4
50	56
60	68
70	77
80	88
90	100

Dari data tersebut maka didapatkan grafik sebagai berikut :



Gambar 4.6 Grafik respon aktuator

#### 4.5 Pengujian Alat

Sensor *flowmeter* pengujian sensor dilakukan dengan memberikan flowrate dari tangki reaktor biogas serta dari tangki storage bensin . Pada Pengendalian rasio kali ini flow bensin

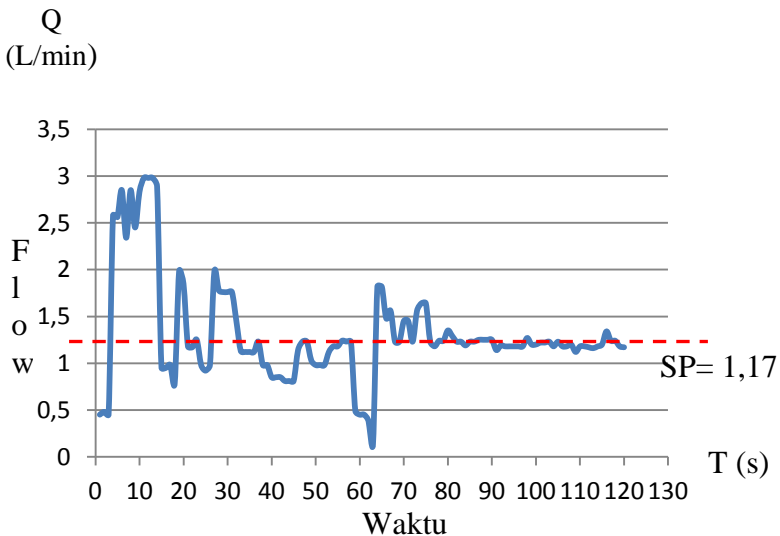
diasumsikan konstan pada flow 0,13 L/min. Hasil pengendalian flowrate biogas dapat dilihat seperti pada tabel 4.1. pengambilan data dilakukan setiap 5 detik. Aktuator motor servo operated valve dapat memberikan output mendekati setpoint sensor sebesar 1,17 L/min. Pengujian dilakukan untuk mengetahui grafik respon aktuator dan pembacaan flowrate pada sensor berdasarkan waktu yang telah ditentukan.

**Tabel 4.2** Hasil Pengujian Sistem rasio 90 : 10

No	Menit ke-	Detik Ke-	Flow biogas (L/min)	Flow bensin (L/min)
1	1	5	2,56	0,13
2		10	2,83	0,13
3		15	0,95	0,13
4		20	1,85	0,13
5		25	0,92	0,13
6		30	1,76	0,13
7		35	1,12	0,13
8		40	0,85	0,13
9		45	0,81	0,13
10		50	0,98	0,13
11		55	0,13	0,13
12		60	0,45	0,13
13	2	65	1,82	0,13
14		70	1,45	0,13
15		75	1,64	0,13
16		80	1,35	0,13
17		85	1,23	0,13
18		90	1,25	0,13
19		95	1,18	0,13
20		100	1,2	0,13

21		105	1,23	0,13
22		110	1,18	0,13
23		115	1,2	0,13
24		120	1.17	0,13

Dari data tersebut maka diperoleh sebuah grafik respon sistem sebagai berikut.



Gambar 4.7 Grafik Hasil Respon system pada rasio 90:10

Dari grafik diatas adalah data flowrate yang di deteksi oleh sensor dengan setpoint 1,17 L/min aliran biogas yang di kendalikan oleh servo operated valve dapat bergerak dengan menyesuaikan setpoint yang telah ditentukan.

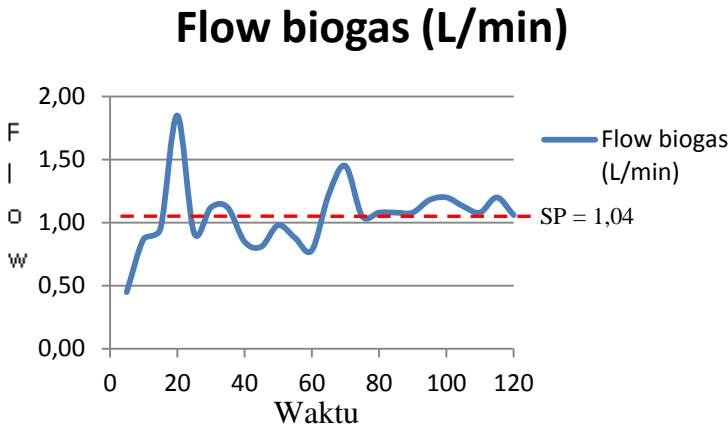
Kemudian pada pengujian kedua ,dilakukan pengambilan data kembali dengan rasio biogas : bensin yaitu 80:20. Dengan flowrate bensin diasumsikan konstan pada 0,26 L/min, dengan

begitu maka seharusnya flowrate biogas harus dijaga set poinnya pada 1,06 L/min

Tabel 4.3 Hasil Pengujian Sistem rasio 80 : 20

No	Menit ke-	Detik Ke-	Flow biogas (L/min)	Flow bensin (L/min)
1	1	5	0,45	0,26
2		10	0,87	0,26
3		15	0,95	0,26
4		20	1,85	0,26
5		25	0,92	0,26
6		30	1,12	0,26
7		35	1,12	0,26
8		40	0,85	0,26
9		45	0,81	0,26
10		50	0,98	0,26
11		55	0,88	0,26
12		60	0,78	0,26
13	2	65	1,22	0,26
14		70	1,45	0,26
15		75	1,05	0,26
16		80	1,08	0,26
17		85	1,08	0,26
18		90	1,08	0,26
19		95	1,18	0,26
20		100	1,2	0,26
21		105	1,13	0,26
22		110	1,08	0,26
23		115	1,2	0,26
24		120	1,06	0,26





Gambar 4. 8 Grafik hasil respon sistem pada rasio 80:20

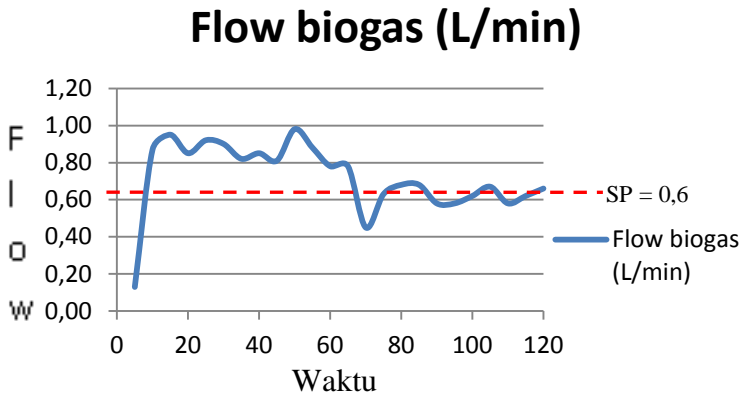
Dari grafik diatas adalah data flowrate yang di deteksi oleh sensor dengan setpoint 1,04 L/min aliran biogas yang di kendalikan oleh servo operated valve dapat bergerak dengan menyesuaikan setpoint yang telah ditentukan.

Dan pengujian ketiga dilakukan kembali dengan rasio biogas : bensin yaitu 70:30. Dengan flowrate bensin diasumsikan konstan pada 0,26 L/min, dengan begitu maka seharusnya flowrate biogas harus dijaga set poinnya pada 0,6 L/min

Tabel 4.4 Hasil Pengujian Sistem rasio 70 : 30

No	Menit ke-	Detik Ke-	Flow biogas (L/min)	Flow bensin (L/min)
1	1	5	0,13	0,26
2		10	0,87	0,26

3		15	0,95	0,26
4		20	0,85	0,26
5		25	0,92	0,26
6		30	0,90	0,26
7		35	0,82	0,26
8		40	0,85	0,26
9		45	0,81	0,26
10		50	0,98	0,26
11		55	0,88	0,26
12		60	0,78	0,26
13	2	65	0,78	0,26
14		70	0,45	0,26
15		75	0,63	0,26
16		80	0,68	0,26
17		85	0,68	0,26
18		90	0,58	0,26
19		95	0,58	0,26
20		100	0,62	0,26
21		105	0,67	0,26
22		110	0,58	0,26
23		115	0,62	0,26
24		120	0,66	0,26



Gambar 4.9 Grafik hasil respon sistem  
pada rasio 70:30

Dari grafik diatas adalah data flowrate yang di deteksi oleh sensor dengan setpoint flowrate biogas 0,6 L/min aliran biogas yang di kendalikan oleh servo operated valve dapat bergerak dengan menyesuaikan setpoint yang telah ditentukan.

#### 4.6 Pembahasan

Tugas akhir yang berjudul **Rancang Bangun Sistem Pengendalian *Flowrate* Bahan Bakar pada Generator Set *Dual Fuel System*** bertujuan untuk mengatur rasio biogas dan bensin yang menjadi bahan bakar generator set dual fuel system. Dengan menggunakan pedoman rasio antara biogas : bensin yaitu 90:10 karena dalam tugas akhir ini ingin menekan penggunaan bensin dan mengoptimalkan penggunaan biogas. Dalam tugas akhir ini menggunakan sensor flowmeter turbin *hall effect* sebagai alat untuk mengukur rasio flowrate pada bensin serta biogas.

Sebelum sensor flowmeter ini digunakan untuk merancang sistem pengendalian maka perlu dilakukan

kalibrasi untuk mengetahui performansi dari sensor tersebut. Kalibrasi sensor flowmeter tersebut menggunakan alat ukur flow yang sudah standar yaitu rotameter. Dilakukan pada range 1-5 L/min. Kedua alat tersebut dialiri udara dengan menggunakan kompresor sehingga dapat diketahui pembacaan rotameter standar dengan flowmeter yang dikalibrasi. Setelah dilakukan pengukuran maka selanjutnya dilakukan perhitungan kalibrasi. Dari perhitungan kalibrasi didapatkan nilai karakteristik statis dan karakteristik dinamis alat dari karakteristik statis dengan range pengukuran 1L/min hingga 5 L/min, didapatkan nilai Span 4 L/min , Akurasi sebesar 59,4%, Histerisis 0,009389671%, Kesalahan (*error*)  $\pm 0,623922025$ . sedangkan dari karakteristik dinamik pengukuran didapatkan bahwa temperature merupakan instrumen orde satu karena saat ada perubahan step input pengukuran, output instrumen berubah secara gradual.

Pada tugas akhir ini menggunakan ATmega128 sebagai kontroler serta CV AVR sebagai software yang digunakan untuk memasukan program pengendalian flowrate . Untuk penyimpanan data menggunakan SD card sehingga mudah untuk membuka hasil pengukuran dan pengendalian flowrate pada biogas serta bensin.

Dalam pengendalian flowrate bahan bakar pada genset dual fuel ini flowmeter digunakan untuk mengukur berapa flowrate antara biogas dan bensin yang masuk ke genset. Hasil pengukuran pada flowmeter ini digunakan sebagai acuan untuk menggerakkan aktuator hingga tercapai set poin dari flow biogas dan bensin. Dalam pengendalian rasio flowrate bahan bakar ini menggunakan rasio 90:10 yaitu bensin 90% dan bensin 10%. dalam tugas akhir ini ingin

menekan penggunaan bensin dan mengoptimalkan penggunaan biogas sehingga rasio biogas jauh lebih besar daripada bensin. Karena nilai flowrate pada bensin ini konstan sehingga kami jaga pada flow 0,13 L/min dan sehingga agar tercapai 90% biogas berarti nilai set poin dari flowrate biogas harus terjaga pada flow 1,17 L/min.

Pada pengujian alat serta pengambilan data diketahui bahwa set poin flowrate dari bensin diasumsikan konstan, dan flowrate biogas sebesar 1,17 L/min. Dalam pengambilan data didapatkan hasil bahwa flowrate biogas paling rendah berada pada 0,13 L/min sedangkan flowrate paling tinggi mencapai 2,83 L/min. Dan waktu yang diperlukan untuk mencapai set poin yaitu pada detik ke-35 dan mencapai steady pada detik ke-100.

Kendala yang ada pada tugas akhir ini yaitu ketersediaan biogas yang masih sulit untuk didapatkan sehingga tidak bisa dilakukan pengujian di lingkungan kampus ITS.

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil Perancangan sistem pengendalian flowrate bahan bakar pada generator set dual fuel system ini didapatkan kesimpulan sebagai berikut

1. Telah dilakukan perancangan pengendalian flowrate bahan bakar biogas dan bensin dirancang dengan rasio perbandingan biogas : bensin yaitu 90:10, 80:20, 70:30
2. Dalam pengujian sistem pengendalian flowrate biogas dan bensin ,pada rasio 90:10 flowrate bensin diasumsikan konstan pada 0,13 L/min sehingga untuk mencapai 90% biogas yaitu flowrate biogas harus pada set poin 1,17 L/min, kemudian pada rasio 80:20 10 flowrate bensin diasumsikan konstan pada 0,26 L/min sehingga untuk mencapai 80% biogas yaitu flowrate biogas harus pada set poin 1,04 L/min, dan pada rasio 70:30 flowrate bensin diasumsikan konstan pada 0,26 L/min sehingga untuk mencapai 70% biogas yaitu flowrate biogas harus pada set poin 0,6 L/min,
3. Telah dilakukan perancangan Air Fuel Rasio pada generator set *dual fuel system*. AFR adalah Air Fuel Rasio yaitu perbandingan antara udara dengan biogas maupun udara dengan bensin , serta udara dengan *dual fuel* (bensin-biogas).
4. Didapat kan nilai Air Fuel Ratio (AFR) udara dan biogas untuk pembakaran sempurna yaitu 1: 3,16 L/min.

#### **5.2 Saran**

Pada Tugas akhir ini terdapat beberapa saran untuk diperbaiki pada Tugas Akhir yang selanjutnya , antara lain :

1. Perlu diperbaiki dalam pemasangan wiring untuk menghindari adanya loss pada wiring yang mengakibatkan supply atau data tidak tersampaikan dengan benar, baik pada kontroler maupun aktuator.
2. Peletakan equipment pada sistem pengendalian perlu diperhatikan kembali sehingga pemasangan dari masing masing equipment pengendalian dapat berfungsi dengan baik.



## DAFTAR PUSTAKA

- Artayana, Ketut Catur Budi. 2014. Pengaruh Variasi Konverter Biogas Terhadap Unjuk Kerja Pada Mesin Genset Berkapasitas 1200 Watt. Denpasar.
- AtMEL. (2005). *Atemega 128/128 L*. Atmel.
- Budiyono, P. M. (2012). Pengaruh metode fermentasi, komposisi umpan, ph awal dan variasi penenceran terhadap produksi biogas dari vinasse. *Jurnal Penelitian Kimia*, 9.2.
- Fahri A. 2011. Teknologi pembuatan biogas dari kotoran ternak.
- Osorio, F. (2009). *Biogas Purification From Anaerobic Digestion In A Waste Water Treatment Plant for Biofuel Production. Renewable Energy*, 34, 2164- 2171.
- Haryati, T. 2006. Biogas: Limbah Peternakan yang Menjadi Sumber Energi Alternatif. *Wartazoa*. 16. 160–169.
- Pertanian, D. J. (2012). *Statistik Peternakan dan Kesehatan Hewan* . Jakarta: Direktorat Jendral Peternakan dan Kesehatan Hewan Kementrian Pertanian RI.
- Rahayu. S., Purwaningsih, D., dan Pujiyanto. 2009. Pemanfaatan Kotoran Ternak Sapi sebagai Sumber Energi Alternatif Ramah Lingkungan Beserta Aspek Sosio Kulturalnya. *Inotek*. 13.2
- Restiya Maulana, M. J. (2016). Karakteristik Flowmeter untuk laju aliran rendah pada sirkulasi alami di untai fassip. *jurnal.umj*, 3.

Statistik Peternakan. 2012. *Populasi ternak sapi potong dan sapi perah tahun 2011*

Wahyuni, S. 2013. *Biogas Energi Alternatif Pengganti BBM, Gas, dan Listrik*. PT Agromedia Pustaka. Jakarta. 117 hlm.

XVII, B. P. (2007). *DASAR INSTRUMENTASI & PROSES KONTROL \_BPST XVII\_*. Balongan: Pertamina.

## LAMPIRAN

**Tabel Hasil Pengambilan Data Naik**

No	Pembacaan standar (L/min)	Data ke					Rata-Rata
		1	2	3	4	5	
1	1	0,88	1,22	1,36	0,94	0,98	1,076
2	2	1,97	2,23	2,26	1,86	2,05	2,074
3	3	3,22	3	3,34	3,34	3,12	3,204
4	4	4,45	4,31	4,45	4,24	4,33	4,356
5	5	5,56	5,34	5,21	5,13	5,43	5,334
Jumlah							16,044

**Tabel Hasil Pengambilan Data Turun**

No	Pembacaan standar (L/min)	Data ke					Rata-Rata
		1	2	3	4	5	
1	1	1,23	1,23	1,36	1,12	1,46	1,28
2	2	1,65	2,22	0,96	2,34	2,32	1,898
3	3	3,24	3,05	3,37	3,33	3,12	3,222
4	4	4,43	4,36	4,22	4,24	4,31	4,312
5	5	5,53	5,32	5,22	5,18	5,43	5,336
Jumlah							16,048

**Tabel Data Kalibrasi Sensor Ua**

No	pembacaan standar	pembacaan alat		rata2	koreksi (Y)
		rata2 naik	rata2 turun		
1	1	1,076	1,28	1,178	-0,178
2	2	2,076	1,898	1,987	0,013
3	3	3,204	3,222	3,213	-0,213
4	4	4,356	4,312	4,334	-0,334
5	5	5,334	5,336	5,335	-0,335
jumlah	15				-1,047
rata - rata	3				-0,2094

Tabel Data Kalibrasi Sensor b

No.	X	X <sup>2</sup>	y-y'	x.y	(y-y') <sup>2</sup>
1	1	1	0,869	-0,178	0,755161
2	2	4	0,2224	0,026	0,04946176
3	3	9	-0,213	-0,639	0,045369
4	4	16	-0,334	-1,336	0,111556
5	5	25	-0,335	-1,675	0,112225
jumlah	15	25	0,2094	-3,802	1,07377276
rata-rata	3	11	0,04188	-0,760	0,112225

Tabel Hasil Data Sensor Naik dan Turun

No	pembacaan standar	pembacaan alat		H	H/pembacaan standar
		rata2 naik	rata2 turun		
1	1	1,076	1,28	0,204	0,204
2	2	2,076	1,898	-0,178	-0,089
3	3	3,204	3,222	0,018	0,006
4	4	4,356	4,312	-0,044	-0,011
5	5	5,334	5,336	0,002	0,0004
jumlah	15	16,046	16,048	0,002	0,1104
rata-rata	3	3,2092	5,349333333	0,0004	0,02208

# Datasheet ATmega 128

## Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 128Kbytes of In-System Self-programmable Flash program memory
  - 4Kbytes EEPROM
  - 4Kbytes Internal SRAM
  - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Up to 64Kbytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V ATmega128L
  - 4.5 - 5.5V ATmega128
- Speed Grades
  - 0 - 8MHz ATmega128L
  - 0 - 16MHz ATmega128



8-bit Atmel  
Microcontroller  
with 128KBytes  
In-System  
Programmable  
Flash

ATmega128  
ATmega128L

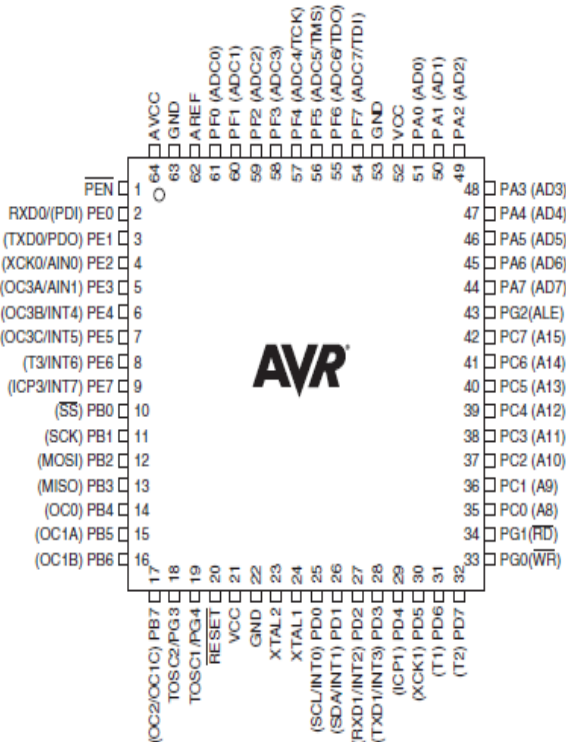
Summary

Rev. 2467XS-AVR-06/11



## Pin Configurations

Figure 1. Pinout ATmega128



Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

## Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Listing Program CV AVR

/\*\*\*\*\*

This program was produced by the

CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 22/06/2018

Author : Biogas Team

Company : Spektron

Comments:

Chip type : ATmega128

Program type : Application

AVR Core Clock frequency: 1,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 1024

\*\*\*\*\*/

```
#include <mega128.h>
```

```
#include <delay.h>
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
// Alphanumeric LCD functions
```

```
#include <alcd.h>
```

```
#define up PIND.4
```

```
#define down PIND.5
```

```
#define start PIND.6
```

```
#define stop PIND.7
```

```
#define off 0
```

```
#define on 1
```

```
int data=0,data1=0,data2=0;
```

```
bit status_up=0,status_down=0;
```

```
int iservo, iservo1, iservo2, iservo3, awal, akhir, setawalservo;
```

```

int ulang,ulang1, ulang2;
char tampil servo[10], tampil servo1[10], tampil servo2[10];
int delay = 0;
    int rev=0;

```

```

void ketikdata() {
    if (up==off) status_up=on;
    if ((up==on)&(status_up==on))
    { data=data+5;
    if (data>=100){ data=100;}
    status_up=off;
    lcd_clear();
    }
    if (down==off) status_down=on;
    if ((down==on)&(status_down==on))
    {
    data=data-5;
    if(data<0){data=0;}
    status_down=off;
    lcd_clear();
    }
}

```

```

void aturser1()
{

```

```

    if (ulang == 0)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1000);
            PORTA.0=0;
            delay_us(19000);
        }
        delay_ms(100);
    }

```

```

    if (ulang == 5)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1035);

```



```

        PORTA.0=0;
        delay_us(18965);
    }
    delay_ms(100);
}

if (ulang == 10)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1070);
        PORTA.0=0;
        delay_us(19930);
    }
    delay_ms(100);
}

if (ulang == 15)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1105);
        PORTA.0=0;
        delay_us(18950);
    }
    delay_ms(100);
}

if (ulang == 20)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1140);
        PORTA.0=0;
        delay_us(18860);
    }
    delay_ms(100);
}

if (ulang == 25)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1175);
    }
}

```

```

PORTA.0=0;
    delay_us(18825);
}
delay_ms(100);
}

if (ulang == 30)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1210);
        PORTA.0=0;
        delay_us(18790);
    }
    delay_ms(100);
}

    if (ulang == 35)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1245);
        PORTA.0=0;
        delay_us(18755);
    }
    delay_ms(100);
}

if (ulang == 40)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1280);
        PORTA.0=0;
        delay_us(18720);
    }
    delay_ms(100);
}

if (ulang == 45)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1315);
    }
}

```

```
    PORTA.0=0;
    delay_us(18685);
  }
  delay_ms(100);
}
```

```
if (ulang == 50)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1350);
    PORTA.0=0;
    delay_us(18650);
  }
  delay_ms(100);
}
```

```
if (ulang == 55)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1385);
    PORTA.0=0;
    delay_us(18615);
  }
  delay_ms(100);
}
```

```
if (ulang == 60)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1420);
    PORTA.0=0;
    delay_us(18580);
  }
  delay_ms(100);
}
```

```
if (ulang == 65)
```

```

{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1455);
        PORTA.0=0;
        delay_us(18545);
    }
    delay_ms(100);
}

else if (ulang == 70)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1490);
        PORTA.0=0;
        delay_us(18510);
    }
    delay_ms(100);
}

if (ulang == 75)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1525);
        PORTA.0=0;
        delay_us(18475);
    }
    delay_ms(100);
}

if (ulang == 80)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1560);
        PORTA.0=0;
        delay_us(18440);
    }
    delay_ms(100);
}

```

```
}
```

```
if (ulang == 85)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1595);
        PORTA.0=0;
        delay_us(18405);
    }
    delay_ms(100);
}
```

```
if (ulang == 90)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1630);
        PORTA.0=0;
        delay_us(18370);
    }
    delay_ms(100);
}
```

```
if (ulang == 95)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1665);
        PORTA.0=0;
        delay_us(18335);
    }
    delay_ms(100);
}
```

```
if (data == 100)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1700);
    }
}
```

```

        PORTA.0=0;
        delay_us(18300);
    }
    delay_ms(100);
}

}

```

```

void aturser2()
{

```

```

    if (ulang1 == 0)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1000);
            PORTA.1=0;
            delay_us(19000);
        }
        delay_ms(100);
    }

```

```

    if (ulang1 == 5)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1035);
            PORTA.1=0;
            delay_us(18965);
        }
        delay_ms(100);
    }

```

```

    if (ulang1 == 10)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1070);
            PORTA.1=0;
            delay_us(19930);
        }
        delay_ms(100);
    }

```

```

if (ulang1 == 15)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1105);
        PORTA.1=0;
        delay_us(18950);
    }
    delay_ms(100);
}

if (ulang1 == 20)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1140);
        PORTA.1=0;
        delay_us(18860);
    }
    delay_ms(100);
}

if (ulang1 == 25)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1175);
        PORTA.1=0;
        delay_us(18825);
    }
    delay_ms(100);
}

if (ulang1 == 30)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1210);
        PORTA.1=0;
        delay_us(18790);
    }
    delay_ms(100);
}

```

```
        if (ulang1 == 35)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1245);
            PORTA.1=0;
            delay_us(18755);
        }
        delay_ms(100);
    }
```

```
    if (ulang1 == 40)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1280);
            PORTA.1=0;
            delay_us(18720);
        }
        delay_ms(100);
    }
```

```
    if (ulang1 == 45)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1315);
            PORTA.1=0;
            delay_us(18685);
        }
        delay_ms(100);
    }
```

```
    if (ulang1 == 50)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
            PORTA.1=1;
            delay_us(1350);
            PORTA.1=0;
            delay_us(18650);
        }
        delay_ms(100);
    }
```



```
if (ulang1 == 55)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1385);
        PORTA.1=0;
        delay_us(18615);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 60)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1420);
        PORTA.1=0;
        delay_us(18580);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 65)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1455);
        PORTA.1=0;
        delay_us(18545);
    }
    delay_ms(100);
}
```

```
else if (ulang1 == 70)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1490);
        PORTA.1=0;
```

```
        delay_us(18510);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 75)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1525);
        PORTA.1=0;
        delay_us(18475);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 80)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1560);
        PORTA.1=0;
        delay_us(18440);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 85)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1595);
        PORTA.1=0;
        delay_us(18405);
    }
    delay_ms(100);
}
```

```
if (ulang1 == 90)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
```

```

        PORTA.1=1;
        delay_us(1630);
        PORTA.1=0;
        delay_us(18370);
    }
    delay_ms(100);
}

```

```

if (ulang1 == 95)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1665);
        PORTA.1=0;
        delay_us(18335);
    }
    delay_ms(100);
}

```

```

if (ulang1 == 100)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1700);
        PORTA.1=0;
        delay_us(18300);
    }
    delay_ms(100);
}

```

```

}

```

```

void aturser3()
{
    data2=100-data2;
    ulang2=7*data2+1000;
    for(iservo3=0;iservo3<=200;iservo3++) { //Bukaservo3
        PORTA.2=1;
        delay_us(1385);
        PORTA.2=0;
        delay_us(18615);
    }
}

```

```
    }  
    delay_ms(100);  
}
```

```
void statusdataservo() {  
    data = data;  
    data1 = 100 - data;  
    data2 = 50;  
}
```

```
void lcdservo() {  
    lcd_gotoxy(0,0);  
    lcd_putsf("Setting Status : ");  
    itoa(data,tampilservo);  
    itoa(data1,tampilservo1);  
    itoa(data2,tampilservo2);  
    lcd_gotoxy(10,1);  
    lcd_puts(tampilservo);  
    lcd_gotoxy(10,2);  
    lcd_puts(tampilservo1);  
    lcd_gotoxy(10,3);  
    lcd_puts(tampilservo2);  
    lcd_gotoxy(0,1);  
    lcd_putsf("Servo 1 : ");  
    lcd_gotoxy(0,2);  
    lcd_putsf("Servo 2 : ");  
    lcd_gotoxy(0,3);  
    lcd_putsf("Servo 3 : ");  
    lcd_gotoxy(13,1);  
    lcd_putsf(" % ");  
    lcd_gotoxy(13,2);  
    lcd_putsf(" % ");  
    lcd_gotoxy(13,3);  
    lcd_putsf(" % ");  
}
```

```
void dataservo() {  
    lcd_gotoxy(4,1);  
    lcd_putsf(" ");  
    lcd_gotoxy(4,2);  
    lcd_putsf(" ");  
    lcd_gotoxy(4,3);
```

```

    lcd_putsf(" ");
    data=data;
    data1=data1;
    data2=data2;
    itoa(data,tampilservo);
    itoa(data1,tampilservo1);
    itoa(data2,tampilservo2);
    lcd_gotoxy(3,1);
    lcd_puts(tampilservo);
    lcd_gotoxy(3,2);
    lcd_puts(tampilservo1);
    lcd_gotoxy(3,3);
    lcd_puts(tampilservo2);
    lcd_gotoxy(0,1);
    lcd_putsf("Sb:");
    lcd_gotoxy(0,2);
    lcd_putsf("Sg:");
    lcd_gotoxy(0,3);
    lcd_putsf("Su:");
    lcd_gotoxy(7,1);
    lcd_putsf("%");
    lcd_gotoxy(7,2);
    lcd_putsf("%");
    lcd_gotoxy(7,3);
    lcd_putsf("%");
}

void my_delay_ms(int us) {
    unsigned char n;
    n= us;

    while(n--)
    {
        delay_us(10);
    }
}

void setservo() {
    data=100-data;
    ulang=(6*data+1000);
    rev = (20000-ulang);

    if (data == 0)
    {   for(iservo1=0;iservo1<=200;iservo1++)

```

```

    {
        PORTA.0=1;
        delay_us(1000);
        PORTA.0=0;
        delay_us(19000);

        PORTA.2=1;
        delay_us(1000);
        PORTA.2=0;
        delay_us(19000);
    }
    delay_ms(100);
}

if (data == 5)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1035);
        PORTA.0=0;
        delay_us(18965);
    }
    delay_ms(100);
}

if (data == 10)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1070);
        PORTA.0=0;
        delay_us(19930);
    }
    delay_ms(100);
}

if (data == 15)
{
    for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1105);
        PORTA.0=0;
        delay_us(18950);
    }
    delay_ms(100);
}

```

```

    }

    if (data == 20)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1140);
            PORTA.0=0;
            delay_us(18860);
        }
        delay_ms(100);
    }

    if (data == 25)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1175);
            PORTA.0=0;
            delay_us(18825);
        }
        delay_ms(100);
    }

    if (data == 30)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1210);
            PORTA.0=0;
            delay_us(18790);
        }
        delay_ms(100);
    }

    if (data == 35)
    {   for(iservo1=0;iservo1<=200;iservo1++)
        {
            PORTA.0=1;
            delay_us(1245);
            PORTA.0=0;
            delay_us(18755);
        }
        delay_ms(100);
    }

```

```
}

if (data == 40)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1280);
        PORTA.0=0;
        delay_us(18720);
    }
    delay_ms(100);
}
```

```
if (data == 45)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1315);
        PORTA.0=0;
        delay_us(18685);
    }
    delay_ms(100);
}
```

```
if (data == 50)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1350);
        PORTA.0=0;
        delay_us(18650);
    }
    delay_ms(100);
}
```

```
if (data == 55)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1385);
        PORTA.0=0;
        delay_us(18615);
    }
}
```



```
    delay_ms(100);  
}
```

```
if (data == 60)  
{   for(iservo1=0;iservo1<=200;iservo1++)  
    {  
        PORTA.0=1;  
        delay_us(1420);  
        PORTA.0=0;  
        delay_us(18580);  
    }  
    delay_ms(100);  
}
```

```
if (data == 65)  
{   for(iservo1=0;iservo1<=200;iservo1++)  
    {  
        PORTA.0=1;  
        delay_us(1455);  
        PORTA.0=0;  
        delay_us(18545);  
    }  
    delay_ms(100);  
}
```

```
else if (data == 70)  
{   for(iservo1=0;iservo1<=200;iservo1++)  
    {  
        PORTA.0=1;  
        delay_us(1490);  
        PORTA.0=0;  
        delay_us(18510);  
    }  
    delay_ms(100);  
}
```

```
if (data == 75)  
{   for(iservo1=0;iservo1<=200;iservo1++)  
    {  
        PORTA.0=1;
```

```
    delay_us(1525);
    PORTA.0=0;
    delay_us(18475);
  }
  delay_ms(100);
}
```

```
if (data == 80)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1560);
    PORTA.0=0;
    delay_us(18440);
  }
  delay_ms(100);
}
```

```
if (data == 85)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1595);
    PORTA.0=0;
    delay_us(18405);
  }
  delay_ms(100);
}
```

```
if (data == 90)
{  for(iservo1=0;iservo1<=200;iservo1++)
  {
    PORTA.0=1;
    delay_us(1630);
    PORTA.0=0;
    delay_us(18370);
  }
  delay_ms(100);
}
```

```
if (data == 95)
```

```

{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1665);
        PORTA.0=0;
        delay_us(18335);
    }
    delay_ms(100);
}

```

```

if (data == 100)
{   for(iservo1=0;iservo1<=200;iservo1++)
    {
        PORTA.0=1;
        delay_us(1700);
        PORTA.0=0;
        delay_us(18300);
    }
    delay_ms(100);
}

```

```

data1=100-data1;
ulang1=7*data1+1000;

```

```

if (data1 == 0)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1000);
        PORTA.1=0;
        delay_us(19000);
    }
    delay_ms(100);
}

```

```

if (data1 == 5)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1035);
    }
}

```

```

        PORTA.1=0;
        delay_us(18965);
    }
    delay_ms(100);
}

if (data1 == 10)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1070);
        PORTA.1=0;
        delay_us(19930);
    }
    delay_ms(100);
}

if (data1 == 15)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1105);
        PORTA.1=0;
        delay_us(18950);
    }
    delay_ms(100);
}

if (data1 == 20)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1140);
        PORTA.1=0;
        delay_us(18860);
    }
    delay_ms(100);
}

if (data1 == 25)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1175);
        PORTA.1=0;
    }
}

```

```

        delay_us(18825);
    }
    delay_ms(100);
}

if (data1 == 30)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1210);
        PORTA.1=0;
        delay_us(18790);
    }
    delay_ms(100);
}

    if (data1 == 35)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1245);
        PORTA.1=0;
        delay_us(18755);
    }
    delay_ms(100);
}

if (data1 == 40)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1280);
        PORTA.1=0;
        delay_us(18720);
    }
    delay_ms(100);
}

if (data1 == 45)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1315);
        PORTA.1=0;
    }
}

```

```
        delay_us(18685);
    }
    delay_ms(100);
}
```

```
if (data1 == 50)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1350);
        PORTA.1=0;
        delay_us(18650);
    }
    delay_ms(100);
}
```

```
if (data1 == 55)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1385);
        PORTA.1=0;
        delay_us(18615);
    }
    delay_ms(100);
}
```

```
if (data1 == 60)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1420);
        PORTA.1=0;
        delay_us(18580);
    }
    delay_ms(100);
}
```

```
if (data1 == 65)
{   for(iservo2=0;iservo2<=200;iservo2++)
    {
```

```
    PORTA.1=1;
    delay_us(1455);
    PORTA.1=0;
    delay_us(18545);
  }
  delay_ms(100);
}
```

```
else if (data1 == 70)
{  for(iservo2=0;iservo2<=200;iservo2++)
  {
    PORTA.1=1;
    delay_us(1490);
    PORTA.1=0;
    delay_us(18510);
  }
  delay_ms(100);
}
```

```
if (data1 == 75)
{  for(iservo2=0;iservo2<=200;iservo2++)
  {
    PORTA.1=1;
    delay_us(1525);
    PORTA.1=0;
    delay_us(18475);
  }
  delay_ms(100);
}
```

```
if (data1 == 80)
{  for(iservo2=0;iservo2<=200;iservo2++)
  {
    PORTA.1=1;
    delay_us(1560);
    PORTA.1=0;
    delay_us(18440);
  }
  delay_ms(100);
}
```

```
if (data1 == 85)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1595);
        PORTA.1=0;
        delay_us(18405);
    }
    delay_ms(100);
}
```

```
if (data1 == 90)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1630);
        PORTA.1=0;
        delay_us(18370);
    }
    delay_ms(100);
}
```

```
if (data1 == 95)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1665);
        PORTA.1=0;
        delay_us(18335);
    }
    delay_ms(100);
}
```

```
if (data1 == 100)
{
    for(iservo2=0;iservo2<=200;iservo2++)
    {
        PORTA.1=1;
        delay_us(1700);
        PORTA.1=0;
        delay_us(18300);
    }
}
```



```

        delay_ms(100);
    }

    data2=100-data2;
    ulang2=7*data2+1000;

    if (data2 == 50)
    {   for(iservo2=0;iservo2<=200;iservo2++)
        {
//          PORTA.2=1;
//          delay_us(1350);
//          PORTA.2=0;
//          delay_us(18650);

          PORTA.2=1;
          delay_us(1000);
          PORTA.2=0;
          delay_us(19000);
        }
        delay_ms(100);
    }

    delay_ms(100);
}

```

```

//Ultrasonic
unsigned int countUS;
float level_cm;
//float waktu_second;
char buff[50];
//int timeOut;
//const int maxTimeOut = 50;
void bacaSensorUS()
{
    countUS = 0;
    DDRB.0 = 1;
    PORTB.0 = 1;
    delay_us(50);
    PORTB.0 = 0;
    while(PINB.1==0)
    {
        //if(timeOut>=maxTimeOut) break;
    }
}

```

```

    };
    while(PINB.1==1)
    {
        //if(timeOut>=maxTimeOut) break;
        countUS++;
    }
    //waktu_second = (countUS*0.000000083);
    level_cm = (((countUS*0.344)/2)-1);
    if (level_cm<0) {level_cm = level_cm*-1;}
    level_cm = level_cm-2;
    level_cm = 30-level_cm;
    if(level_cm>=30) level_cm = 30;
}

void dataultrasonic(){
    bacaSensorUS();
    sprintf(buff,"L:%.2f cm",level_cm);
    lcd_gotoxy(0,0);
    lcd_puts(buff);
    delay_ms(100);
}

//Flowmeter
float freqflow, freqflow1, freqflow2; // to store value of frequency
value
unsigned int iflow=0, iflow1=0, iflow2=0; //i=number of overflows in
one second
char bufferflow[8], bufferflow1[8], bufferflow2[8]; // to store the
frequency value as a string to be displayed on lcd

void dataflow() {
    iflow=0;
    iflow1=0;
    iflow2=0;
    delay_ms(100);
    freqflow = ((iflow * 2.25)*60)/1000;
    freqflow1 = ((iflow1 * 2.25)*60)/1000;
    freqflow2 = ((iflow2 * 2.25)*60)/1000;
    if (freqflow >=30 ) {freqflow = 30;}
    if (freqflow1 >=30 ) {freqflow1 = 30;}
    if (freqflow2 >=30 ) {freqflow2 = 30;}
    lcd_gotoxy(10,1);
    lcd_putsf("Qb:");
    lcd_gotoxy(10,2);
    lcd_putsf("Qg:");
}

```

```

    lcd_gotoxy(10,3);
    lcd_putsf("Qu:");
    ftoa(freqflow,2,bufferflow);
    ftoa(freqflow1,2,bufferflow1);
    ftoa(freqflow2,2,bufferflow2);
    lcd_gotoxy(13,1);
    lcd_puts(bufferflow);
    lcd_gotoxy(13,2);
    lcd_puts(bufferflow1);
    lcd_gotoxy(13,3);
    lcd_puts(bufferflow2);
    lcd_gotoxy(17,1);
    lcd_putsf("L/m");
    lcd_gotoxy(17,2);
    lcd_putsf("L/m");
    lcd_gotoxy(17,3);
    lcd_putsf("L/m");
    delay_ms(100);
}

//Optocoupler
int frekuensiop=0, pulsaop, speedop;
char op[10];
unsigned char counterop=0;

//Otomatis Servo

float flowservo, flowservo1, flowservo2;
float freqflowc, freqflowc1, freqflowc2;
float flowservodat, flowservodat1, flowservodat2;

void range1()
{
    if (flowservodat<5)
    {
        flowservodat=0;
    }
    if (flowservodat>=5&&flowservodat<10)
    {
        flowservodat=5;
    }
    if (flowservodat>=10&&flowservodat<15)
    {
        flowservodat=10;
    }
}

```

```
}
  if (flowservodat>=15&&flowservodat<20)
  {
    flowservodat=15;
  }
  if (flowservodat>=20&&flowservodat<25)
  {
    flowservodat=20;
  }
  if (flowservodat>=25&&flowservodat<30)
  {
    flowservodat=25;
  }
  if (flowservodat>=30&&flowservodat<35)
  {
    flowservodat=30;
  }
  if (flowservodat>=35&&flowservodat<40)
  {
    flowservodat=35;
  }
  if (flowservodat>=40&&flowservodat<45)
  {
    flowservodat=40;
  }
  if (flowservodat>=45&&flowservodat<50)
  {
    flowservodat=45;
  }
  if (flowservodat>=50&&flowservodat<55)
  {
    flowservodat=50;
  }
  if (flowservodat>=55&&flowservodat<60)
  {
    flowservodat=55;
  }
  if (flowservodat>=60&&flowservodat<65)
  {
    flowservodat=60;
  }
  if (flowservodat>=65&&flowservodat<70)
  {
    flowservodat=65;
  }
}
```

```

        if (flowservodat>=70&&flowservodat<75)
        {
            flowservodat=70;
        }
        if (flowservodat>=75&&flowservodat<80)
        {
            flowservodat=75;
        }
        if (flowservodat>=80&&flowservodat<85)
        {
            flowservodat=80;
        }
        if (flowservodat>=85&&flowservodat<90)
        {
            flowservodat=85;
        }
        if (flowservodat>=90&&flowservodat<95)
        {
            flowservodat=90;
        }
        if (flowservodat>=95&&flowservodat<100)
        {
            flowservodat=95;
        }
    }

```

```

void range2()
{
    if (flowservodat1<5)
    {
        flowservodat1=0;
    }
    if (flowservodat1>=5&&flowservodat1<10)
    {
        flowservodat1=5;
    }
    if (flowservodat1>=10&&flowservodat1<15)
    {
        flowservodat1=10;
    }
    if (flowservodat1>=15&&flowservodat1<20)
    {
        flowservodat1=15;
    }
}

```

```
}
  if (flowservodat1>=20&&flowservodat1<25)
  {
    flowservodat1=20;
  }
  if (flowservodat1>=25&&flowservodat1<30)
  {
    flowservodat1=25;
  }
  if (flowservodat1>=30&&flowservodat1<35)
  {
    flowservodat1=30;
  }
  if (flowservodat1>=35&&flowservodat1<40)
  {
    flowservodat1=35;
  }
  if (flowservodat1>=40&&flowservodat1<45)
  {
    flowservodat1=40;
  }
  if (flowservodat1>=45&&flowservodat1<50)
  {
    flowservodat1=45;
  }
  if (flowservodat1>=50&&flowservodat1<55)
  {
    flowservodat1=50;
  }
  if (flowservodat1>=55&&flowservodat1<60)
  {
    flowservodat1=55;
  }
  if (flowservodat1>=60&&flowservodat1<65)
  {
    flowservodat1=60;
  }
  if (flowservodat1>=65&&flowservodat1<70)
  {
    flowservodat1=65;
  }
  if (flowservodat1>=70&&flowservodat1<75)
  {
    flowservodat1=70;
  }
}
```

```

    if (flowservodat1>=75&&flowservodat1<80)
    {
        flowservodat1=75;
    }
    if (flowservodat1>=80&&flowservodat1<85)
    {
        flowservodat1=80;
    }
    if (flowservodat1>=85&&flowservodat1<90)
    {
        flowservodat1=85;
    }
    if (flowservodat1>=90&&flowservodat1<95)
    {
        flowservodat1=90;
    }
    if (flowservodat1>=95&&flowservodat1<100)
    {
        flowservodat1=95;
    }
}

```

```

void otoservo() {
    if (freqflow == freqflowc) {ulang=0;}
    if (freqflow != freqflowc) {
        flowservo = (freqflow/6)*100;
        if (flowservo > data) {
            flowservodat = flowservo - data ;
            rangel();
            ulang=flowservodat;

            aturserl();

            delay_ms(100);
            data = flowservo;
        }
    }
}

```

```

if (freqflow == 0) {
    flowservo = 100;
    flowservodat = flowservo - data;
    rangel();
    ulang=flowservodat ;
    aturserl();
    delay_ms(100);
}

```

```

        data = flowservo;
    }

    if (flowservo < data) {
        flowservodat = data - flowservo;
        range1();
        ulang=flowservodat;
        aturser1();

        delay_ms(100);
        data = flowservo;
    }
}

freqflowc = freqflow;

if (freqflow1 == freqflowc1) {ulang1=0;}
if (freqflow1 != freqflowc1) {
    flowservo1 = (freqflow1/6)*100;
    if (flowservo1 > data1) {
        flowservodat1 = flowservo1 - data1;
        range2();
        ulang1=flowservodat1;

        aturser2();

        delay_ms(100);
        data1 = flowservo1;
    }

    if (freqflow1 == 0) {
        flowservo1 = 100;
        flowservodat1 = flowservo1 - data1;
        range2();
        ulang1=flowservodat1;
        aturser2();
        delay_ms(100);
        data1 = flowservo1;
    }

    if (flowservo1 < data1) {
        flowservodat1 = data1 - flowservo1;
        range2();
        ulang1=flowservodat1;
    }
}

```



```

        aturser2();
        delay_ms(100);
        data1 = flowservo1;
    }

}

freqflowc1 = freqflow1;

if (freqflow2 == freqflowc2) {ulang2=0;}
if (freqflow2 != freqflowc2) {
    flowservo2 = (freqflow2/30)*100;
    if (flowservo2 > data2) {
        flowservodat2 = flowservo2 - data2;
        ulang2=(flowservodat2*0.24)+8;
        for(iservo3=0;iservo3<=200;iservo3++) { //Buka Servo 3
            PORTA.2=1;
            delay_us(1000);
            PORTA.2=0;
            delay_us(19000);
        }
        delay_ms(100);
        data2 = flowservo2;
    }

    if (freqflow2 == 0) {
        flowservo2 = 100;
        flowservodat2 = flowservo2 - data2;
        ulang2=(flowservodat2*0.24)+8;
        for(iservo3=0;iservo3<=ulang2;iservo3++) { //Buka Servo 3
            PORTA.2=1;
            delay_us(1000);
            PORTA.2=0;
            delay_us(19000);
        }
        delay_ms(100);
        data2 = flowservo2;
    }

    if (flowservo2 < data2) {
        flowservodat2 = data2 - flowservo2;
        ulang2=(flowservodat2*0.35)+8;
        for(iservo3=0;iservo3<=ulang2;iservo3++) { //Tutup Servo 3
            PORTA.2=1;

```

```

        delay_us(1000);
        PORTA.2=0;
        delay_us(19000);
    }
    delay_ms(100);
    data2 = flowservo2;
}

}

freqflowc2 = freqflow2;
}

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    frekuensiop++;
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    // Place your code here
}

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
    // Place your code here
    iflow++;
}

// External Interrupt 5 service routine
interrupt [EXT_INT5] void ext_int5_isr(void)
{
    // Place your code here
    iflow1++;
}

// External Interrupt 6 service routine

```

```

interrupt [EXT_INT6] void ext_int6_isr(void)
{
    // Place your code here
    iflow2++;
}

```

```

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Reinitialize Timer 0 value
    TCNT0=0xB2;
    // Place your code here
    counterop++;
    if (counterop==15) {
        pulsaop=((float)frekuensiop*60);
        speedop=pulsaop;
        counterop=0;
        pulsaop=0;
        frekuensiop=0;
    }
}

```

```

void datarpm() {
    lcd_gotoxy(11,0);
    lcd_putsf("RPM:");
    lcd_gotoxy(16,0);
    lcd_putsf(" ");
    itoa(speedop,op);
    lcd_gotoxy(15,0);
    lcd_puts(op);
    delay_ms(100);
}

```

```

// Timer2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
    // Reinitialize Timer2 value
    TCNT2=0xB2;
    // Place your code here

}
int a=3;
void simpan_data()

```

```

{
    if(xcount==2)
    {
        printf("Level   = %.2f cm ",level);
        putchar(10);
        putchar(13);
        putchar(10);
        putchar(13);
        printf("Flow Bensin = %.2f L/Menit ",freqflow);
        putchar(10);
        putchar(13);
        putchar(10);
        putchar(13);
        printf("Flow Biogas   = %.2f L/Menit ",freqflow1);
        putchar(10);
        putchar(13);
        putchar(10);
        putchar(13);
        printf("Flow udara     = %.2f L/Menit ",freqflow2);
        putchar(10);
        putchar(13);
        putchar(10);
        putchar(13);
        printf("RPM = &2.f rmp, = &.2f", datarpm);
        putchar(10);
        putchar(13);
        putchar(10);
        putchar(13);
        xcount=0;
    }
}

```

```

// Standard Input/Output functions
#include <stdio.h>

```

```

// Declare your global variables here

```

```

void main(void)
{
    // Declare your local variables here

```

```
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0xFF;
```

```
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x03;
DDRB=0x01;
```

```
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;
```

```
// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTE=0x00;
DDRE=0x00;
```

```
// Port F initialization
```

```

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x06;
TCNT0=0xB2;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;

```

```
OCR1BH=0x00;  
OCR1BL=0x00;  
OCR1CH=0x00;  
OCR1CL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
TCCR2=0x04;  
TCNT2=0xB2;  
OCR2=0x00;
```

```
// Timer/Counter 3 initialization  
// Clock source: System Clock  
// Clock value: Timer3 Stopped  
// Mode: Normal top=0xFFFF  
// OC3A output: Discon.  
// OC3B output: Discon.  
// OC3C output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer3 Overflow Interrupt: Off  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: Off  
// Compare B Match Interrupt: Off  
// Compare C Match Interrupt: Off  
TCCR3A=0x00;  
TCCR3B=0x00;  
TCNT3H=0x00;  
TCNT3L=0x00;  
ICR3H=0x00;  
ICR3L=0x00;  
OCR3AH=0x00;  
OCR3AL=0x00;  
OCR3BH=0x00;  
OCR3BL=0x00;  
OCR3CH=0x00;  
OCR3CL=0x00;
```

```
// External Interrupt(s) initialization  
// INT0: On  
// INT0 Mode: Rising Edge
```

```
// INT1: On
// INT1 Mode: Rising Edge
// INT2: On
// INT2 Mode: Rising Edge
// INT3: On
// INT3 Mode: Rising Edge
// INT4: On
// INT4 Mode: Rising Edge
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x0F;
EICRB=0x3F;
EIMSK=0x73;
EIFR=0x73;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x41;
ETIMSK=0x00;
```

```
// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: Off
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud Rate: 9600
//UCSR0A=0x00;
//UCSR0B=0x08;
//UCSR0C=0x06;
//UBRR0H=0x00;
//UBRR0L=0x06;
```

```
UCSR0A=0x02;
UCSR0B=0x08;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x0C;
// USART1 initialization
// USART1 disabled
UCSR1B=0x00;
```

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
```



```

ACSR=0x80;
SFIO=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);

lcd_init(20);
lcd_gotoxy(0,0);
lcd_putsf("TUGAS AKHIR");
lcd_gotoxy(0,1);
lcd_putsf("BIOGAS");
lcd_gotoxy(0,2);
lcd_putsf("ITS SURABAYA");
delay_ms(100);

if(data<0) {data=0;}
awal=0; akhir=1;

for(iservo=0;iservo<=80;iservo++) { //Kondisi Awal Servo
PORTA.0=1;
PORTA.1=1;
PORTA.2=1;

```

```
delay_us(400);
PORTA.0=0;
PORTA.1=0;
PORTA.2=0;
delay_us(19600);
}
delay_ms(100);
```

```
// Global enable interrupts
#asm("sei")
```

```
while (1)
{
    // Place your code here

    if (awal==0 && akhir==1) {
        ketikdata();
        statusdataservo();
        lcdservo();
        setawalservo=0;
    }

    if (start==on) {
        awal=1;
        akhir=0;
        lcd_clear();
    }

    if (awal==1 && akhir==0) { //DataTampil
        while(1) {
            if (setawalservo==0) {
                setservo();
                setawalservo=1;
            }
        }
        #asm("sei")
        dataservo();
        dataultrasonic();
        dataflow();
        otoservo();
        datarpm();
        usart();
        // printf ("%f\r\n",level_cm);
```

```

        if (stop==on) {
            awal=0;
            akhir=1;
            lcd_clear();
            break;
        }

    }

}

} while (1)
{
    // Place your code here
    dataservo();
    dataultrasonic();
    dataflow();
    otoservo();
    datarpm();
    usart(); ;

    simpan_data();

}
}

```

## BIODATA



Renantyo Ardi Prayogo yang biasa dipanggil “Tyo” merupakan mahasiswa D3 Teknik Instrumentasi Institut Teknologi Sepuluh Nopember. Dilahirkan pada 23 April 1997 di Surabaya. Penulis menempuh pendidikan formal di SDN Tanjung Gresik (2003-2009), SMPN 2 Krian Sidoarjo (2009-2012), SMAN 1 Krian Sidoarjo (2012-2015), dan terakhir menempuh pendidikan

Diploma III Teknik Instrumentasi ITS. Selama kuliah penulis aktif di Workshop Instrumentasi tahun kepengurusan 2016/2017 , 2017/2018. Dan pada akhir kuliahnya penulis telah menyelesaikan Tugas Akhir dengan judul “**Rancang Bangun Sistem Pengendalian Flowrate Bahan Bakar Generator Set *Dual Fuel System***”. Apabila terdapat pertanyaan terkait tugas akhir maka dapat menghubungi penulis di email : tyoardie10@yahoo.com.